Improved whole-chromosome phasing for disease and population genetic studies

To the Editor: Methods that can accurately estimate haplotypes from single-nucleotide polymorphism (SNP) genotype data are important because they are widely used in many areas of genetic analysis. Examples include the creation of haplotype reference panels, pre-phasing[1] before genotype imputation in genome-wide association studies (GWAS), and population genetic analysis. The task is an inverse problem in which we observe a set of SNP genotypes in a sample, typically using a genome-wide SNP microarray, and wish to infer the underlying haplotypes carried by the study individuals.

We recently described the SHAPEIT1 method for inferring haplotype phase from genotype data[2], which improves accuracy and computational efficiency compared to other methods for sample sizes up to ~1,200. Here we present SHAPEIT2, a method that combines features of SHAPEIT1 and Impute2 (ref. 3) to substantially enhance performance. We use the SHAPEIT1 Markov model that represents the space of haplotypes consistent with a given individual's genotypes across a whole chromosome. The transition probabilities of this model are estimated by applying the Impute2 'surrogate family' phasing approach in local windows of size W. In each window, K informative haplotypes are chosen to update the transition probabilities of the Markov model. The method generalizes the Impute2 method so that it can be applied across a whole chromosome with linear computational scaling in K (**Supplementary Methods**). SHAPEIT2 uses multithreading so that multiple cores can be used to phase whole chromosomes, allowing users to make the best use of their computational resources.

We tested SHAPEIT2 on several large-sample, whole-chromosome data sets from a range of SNP genotyping chips (**Supplementary Note 1**). SHAPEIT2 outperforms other methods (**Fig. 1a–c**) in terms of switch error rate (SER) and the mean distance between switch errors (**Supplementary Figs. 1** and **2**). As compared to SHAPEIT1, SHAPEIT2 reduced SER by as much as 45% on these data sets. For example, on 1,229 Vietnamese samples assayed on the Illumina 660K chip on chromosome 22, the SERs of SHAPEIT2 (K = 100, W = 2 Mb), SHAPEIT1 (K = 100) (ref. 2), HAPI-UR (v1.01) (ref. 4), Beagle (v3.3) (ref. 5), Impute2 v2.1.2 (K = 100) (ref. 3), MaCH v1.0.18 (K = 100) (ref. 6) and fastPHASE (v1.4) (ref. 7) were 2.87%, 4.64%, 4.75%, 5.14%, 5.57%, 6.05% and 6.34%, respectively. In general, SHAPEIT2 with low values of K outperformed SHAPEIT1 with high values of K (**Fig. 1a–c**). As the number of samples increased (up to ~9,000 samples in our tests), we found that SHAPEIT2 outperformed other methods and had the property that SER decreases as sample size increases (**Fig. 1d**).

We assessed accuracy on sequence data by phasing 381 European samples from the 1000 Genomes Project (TGP) together with genotypes from two trio parents sequenced at high coverage. We found that SHAPEIT2 (K = 100, W = 0.3 Mb) reduced SER by 38% compared to Beagle (**Supplementary Table 1** and **Supplementary Note 2**), illustrating that the SHAPEIT2 model can adapt to data sets with very high SNP density.

The computational performance of SHAPEIT2 is competitive compared to other methods. **Figure 1e** shows the computational

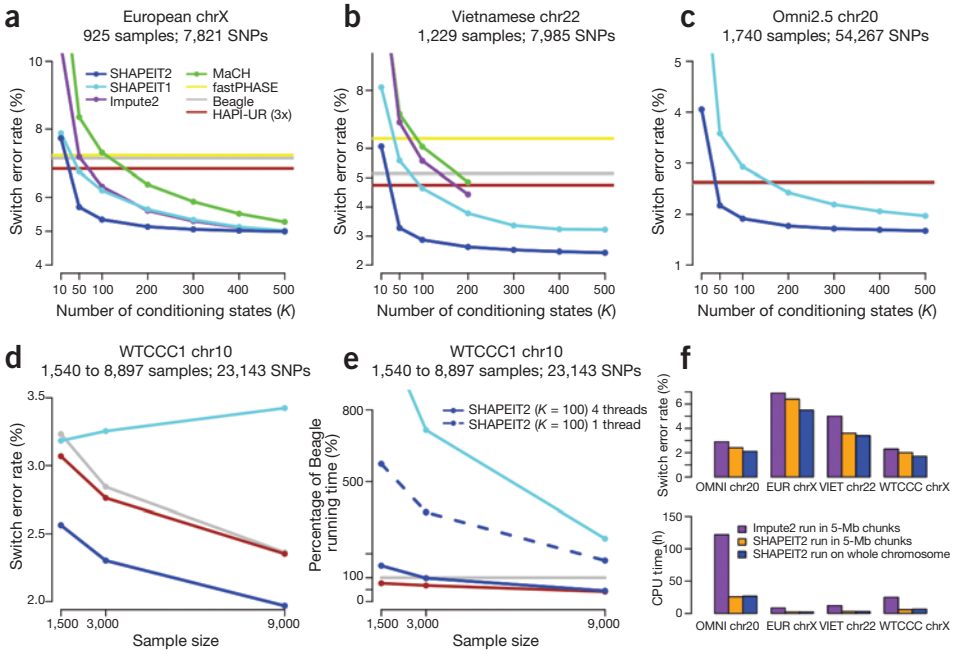


Figure 1 | Accuracy and computational performance. (**a**–**c**) Comparison of methods on (**a**) the European X chromosome data set, (**b**) the Vietnamese chromosome 22 data set and (**c**) the Illumina Omni2.5M chromosome 20 data set. Switch error rate (SER) is plotted against the number of conditioning states (K) for SHAPEIT1 (cyan), SHAPEIT2 with W = 2 Mb (blue), Impute2 (purple) and MaCH (green). fastPHASE (yellow), Beagle (gray) and HAPI-UR (brown) were run using default settings. (**d**) SER versus sample size for SHAPEIT1 (K = 100, W = 2 Mb), SHAPEIT2 (K = 100, W = 2 Mb), Beagle and HAPI-UR on the WTCCC1 data set. (**e**) Running times of the methods relative to Beagle on the WTCCC1 data set. To illustrate benefits of multithreading, SHAPEIT2 was run using one thread (dotted blue line) and four threads (solid blue line). (**f**) SER (top) and CPU time (bottom) of SHAPEIT2 run on a whole chromosome (blue) and of partition-ligation using Impute2 (purple) and SHAPEIT2 (orange).

time (relative to Beagle) of SHAPEIT1 (K = 100), SHAPEIT2 (K = 100) with one and four threads and HAPI-UR (3×) on data sets of up to ~9,000 samples. On a data set of 8,897 samples on chromosome 10, the running times of SHAPEIT2 (K = 100) with four threads, HAPI-UR (3×) and Beagle were 24.3 h, 22.5 h and 52.9 h, respectively. On data sets with high SNP density, computational performance was also good (**Supplementary Table 1**).

An alternative strategy for phasing whole chromosomes, called partition-ligation (PL), involves estimating haplotypes in overlapping windows in parallel, followed by ligation across windows. Running SHAPEIT2 once per chromosome has better accuracy than PL but is much simpler to apply (**Fig. 1f** and **Supplementary Note 2**). It may be possible to improve the performance of PL with alternative strategies.

We investigated the downstream impact of phasing GWAS samples on subsequent imputation performance (**Supplementary Note 2**). We found that SHAPEIT2 haplotypes led to a clear boost in imputation performance compared to other methods (**Supplementary Figs. 3** and **4**).

Large panels of haplotypes from projects such as the TGP might be used to help phase new cohorts, effectively by increasing sample size. This functionality has been included within SHAPEIT2. We phased different-sized subsets of the Vietnamese cohort separately and then together with the TGP Phase 1 haplotypes (**Supplementary Note 2**). Phasing with a reference panel improved performance, but only when the data set had fewer than 100 samples (**Supplementary Fig. 5**). When there is a close match in ancestry between study samples and reference panel, we expect that accuracy will improve with larger study samples.

When a cohort consists of samples with a diverse set of ancestries, one practical question that arises is whether to phase the samples together or separately within distinct ancestral groups. Using 2,123 TGP samples from 14 distinct populations, we found that performance was improved by phasing all samples together rather than separately in continental groups (**Supplementary Figs. 6** and **7**). These results suggest that SHAPEIT2 not only is robust to diverse ancestries but can take advantage of haplotype sharing between populations to improve performance.

SHAPEIT2 is available as **Supplementary Software** and at http://www.shapeit.fr/.

Note: Supplementary information is available at http://www.nature.com/doifinder/10.1038/nmeth.2307.

ACKNOWLEDGMENTS

J.M. and O.D. acknowledge support from the Medical Research Council (G0801823). O.D. acknowledges support from Peptinov SAS (France). Thanks to B. Howie, C. Churchhouse and J. O'Connell for comments on this paper and to A. Cox (Illumina Cambridge Ltd) for providing the high-coverage trio sequence data set. The Vietnamese cohort was provided by A. Alcais (Institut National de la Santé de la Recherche Médicale, Paris, France) and E. Schurr (McGill Centre for the Study of Host Resistance, Montreal, Canada). This study uses data from the Wellcome Trust Case Control Consortium.

AUTHOR CONTRIBUTIONS

O.D. and J.M. designed and performed the research. J.M. and J.-F.Z. supervised the research. J.M. and O.D. wrote the paper.

COMPETING FINANCIAL INTERESTS

The authors declare no competing financial interests.

Olivier Delaneau[1], Jean-Francois Zagury[2,4] & Jonathan Marchini[1,3,4]

[1]Department of Statistics, University of Oxford, Oxford, UK. [2]Chaire de Bioinformatique, Laboratoire Génomique, Bioinformatique, et Applications (EA 4627), Conservatoire National des Arts et Métiers, Paris, France. [3]Wellcome Trust Centre for Human Genetics, University of Oxford, Oxford, UK. [4]These authors contributed equally to this work.
e-mail: marchini@stats.ox.ac.uk or zagury@cnam.fr

1. Howie, B. *et al. Nat. Genet.* **44**, 955–959 (2012).
2. Delaneau, O., Marchini, J. & Zagury, J.F. *Nat. Methods* **9**, 179–181 (2012).
3. Howie, B.N., Marchini, J. & Stephens, M. *G3* **1**, 457–470 (2011).
4. Williams, A.L. *et al. Am. J. Hum. Genet.* **91**, 238–251 (2012).
5. Browning, S.R. & Browning, B.L. *Am. J. Hum. Genet.* **81**, 1084–1097 (2007).
6. Li, Y. *et al. Genet. Epidemiol.* **34**, 816–834 (2010).
7. Scheet, P. & Stephens, M. *Am. J. Hum. Genet.* **78**, 629–644 (2006).

High-resolution whole-genome haplotyping using limited seed data

To the Editor: The term 'haplotype' refers to a group of alleles inherited on the same chromosome. Although most sequencing technologies cannot resolve which chromosomal copy a given sequence comes from, the value of haplotype information for genetic studies is increasingly being appreciated by researchers. Haplotypes are important for inferring disease status, determining which allele combinations tend to segregate together and helping to 'fill in' or impute missing values in regions that lack genotype information to power association studies. Statistical methods exist that can resolve or 'phase' haplotypes, but these have been limited to short sequence stretches and can be computationally demanding. Technical difficulties still exist for obtaining accurate whole-genome and long-range haplotypes experimentally[1,2].

Two high-throughput experimental haplotyping approaches have been developed recently. A single-chromosome isolation approach can yield entire chromosomal haplotypes, but resolution is low because of locus dropout during single-molecule whole-genome amplification[3–5]. A fosmid-based approach yields high-resolution haplotypes but not of chromosome length[6]. To improve the resolution of our previously described single-chromosome approach[4], we have developed the haplotype imputation from incomplete data (HiFi) software. Using limited experimental seed data, HiFi can yield two integral, high-resolution personal chromosomal haplotypes in a cost- and time-efficient manner.

Our aim was to integrate the chromosomal-range accuracy of experimental haplotyping with the efficiency of computational approaches. HiFi exhaustively seeks unambiguous matches to an individual's seed haplotypes and genotypes among a panel of reference haplotypes along a sliding window (**Fig. 1** and **Supplementary Methods**). Once HiFi identifies a single match in a window, it uses the identified reference haplotypes to impute the phases at all loci within this window. If HiFi does not find a unique match, it adjusts the window size and repeats the search automatically.

We examined HiFi performance on three data sets (**Supplementary Table 1**), measuring accuracy as the concordance of HiFi output with high-confidence phase results from trio families (**Supplementary Methods**). We simulated the first data set from HapMap trio haplotypes; then we blinded the phases randomly at 70% of the entire (homozygous and heterozygous) single-nucleotide polymorphism (SNP) set. The second data set contained haplotype-resolved SNPs at ~40.7% of heterozygous loci with the single-chromosome isolation approach[3]. We observed 99.5% (Caucasian) and

Supplementary Information for 'Improved whole chromosome phasing for disease and population genetic studies'.

Olivier Delaneau, Jean-Francois Zagury, Jonathan Marchini

December 4, 2012

Contents

Nature Methods: doi:10.1038/nmeth.2307

Supplementary Information for 'Improved whole chromosome phasing for disease and population genetic studies'.

Olivier Delaneau, Jean-Francois Zagury, Jonathan Marchini

December 4, 2012

Contents

Nature Methods: doi:10.1038/nmeth.2307

Nature Methods: doi:10.1038/nmeth.2307

Nature Methods: doi:10.1038/nmeth.2307

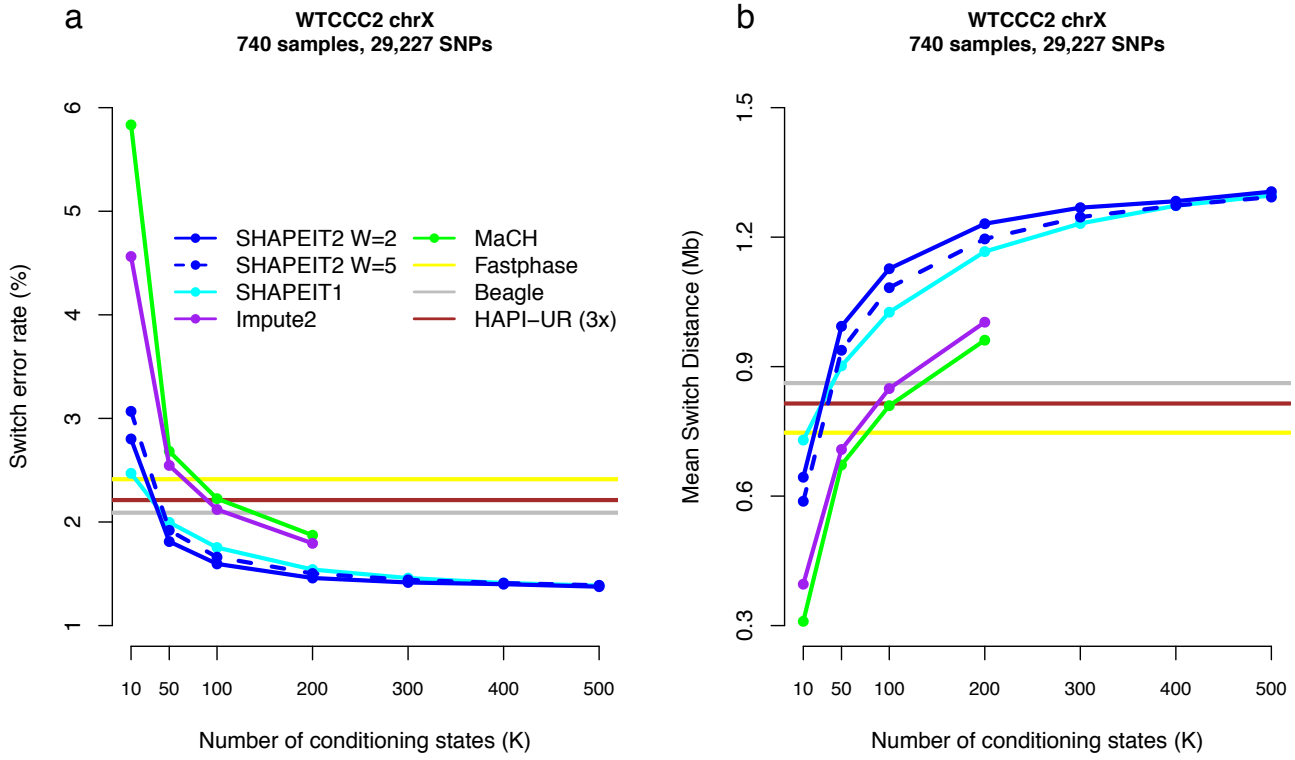
1 Supplementary Table 1

Software	SER (%)	MSD (kb)	Running times (hours)
Beagle	1.72	115	6
SHAPEIT2 W=0.3Mb K=100	1.12	179	5.8
SHAPEIT2 W=0.5Mb K=100	1.16	172	5.75
SHAPEIT2 W=1Mb K=100	1.21	167	5.75
SHAPEIT2 W=2Mb K=100	1.25	161	5.75

Supplementary Table 1 : Methods comparison on high-density SNP genotypes from sequencing Comparison of accuracy and running times of Beagle and SHAPEIT2 on the high-coverage trio parents merged with the 381 individuals of European ancestry of the 1000 Genomes Project. The timings for SHAPEIT2 are based on running the program with 4 threads.

Nature Methods: doi:10.1038/nmeth.2307

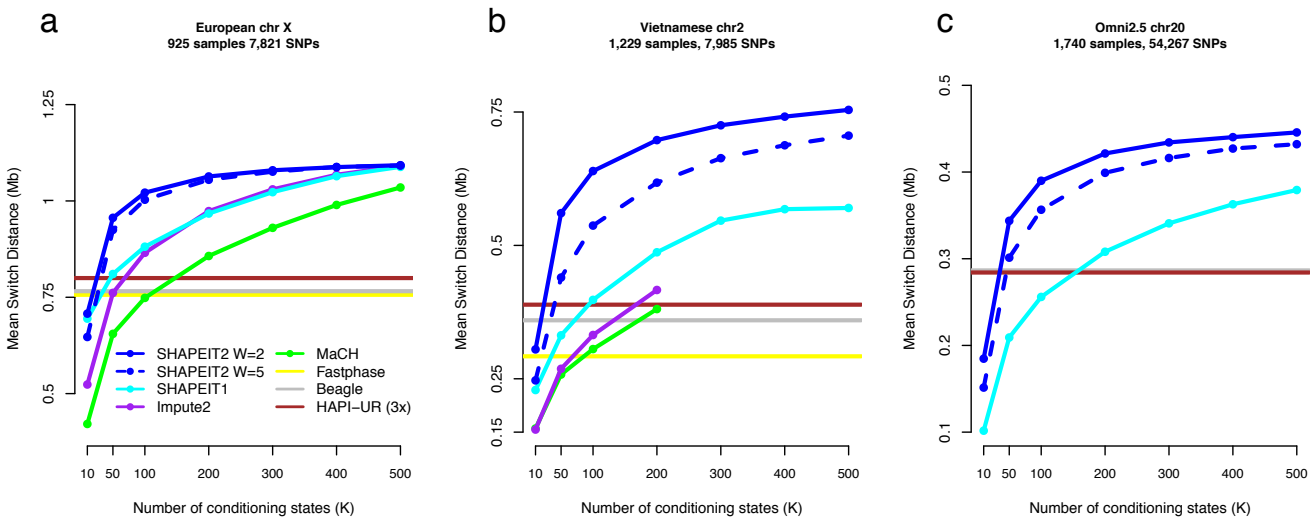
2 Supplementary Figure 1



Supplementary Figure 1 : Comparison of methods on the WTCCC2 X chromosome dataset Switch error rate (a) and mean switch distance in Mb (b) are plotted against the number of conditioning states for SHAPEIT1 (cyan), SHAPEIT2 with W=5Mb (solid blue), SHAPEIT2 with W=2Mb (dashed blue), Impute2 (purple) and MaCH (green). Beagle (grey), HAPI-UR (brown) and Fastphase (yellow) were run using default settings.

Nature Methods: doi:10.1038/nmeth.2307

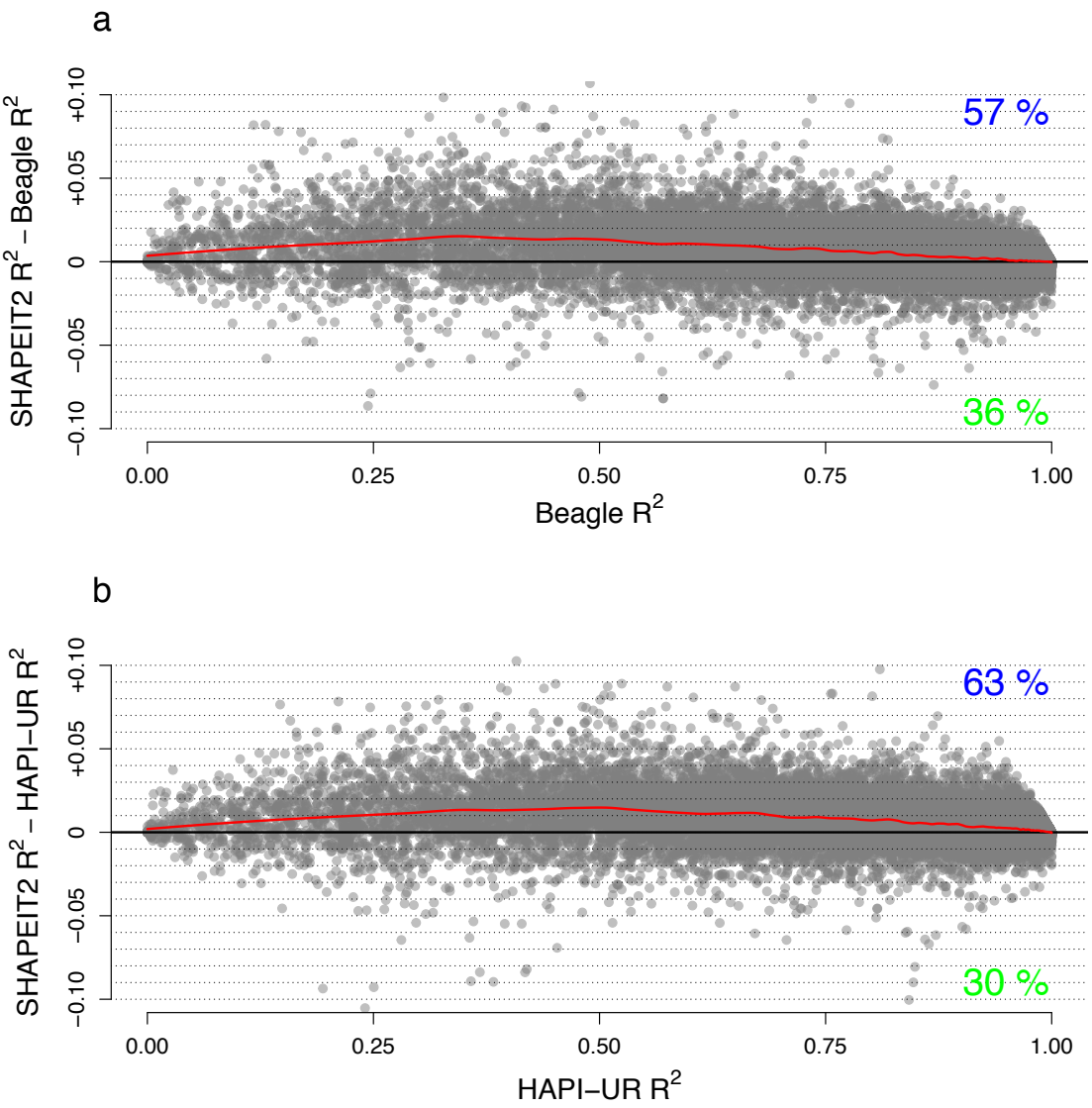
3 Supplementary Figure 2



Supplementary Figure 2 : Comparison of methods measured by Mean Switch Distance European X chromosome dataset (a), WTCCC2 X chromosome dataset (b) and Vietnamese chromosome 22 dataset (c). Mean switch distance in Mb is plotted against the number of conditioning states for SHAPEIT1 (cyan), SHAPEIT2 with 5Mb window (solid blue), SHAPEIT2 with 2Mb window (dashed blue), Impute2 (purple) and MaCH (green). Beagle (grey), HAPI-UR (brown) and Fastphase (yellow) were run using default settings.

Nature Methods: doi:10.1038/nmeth.2307

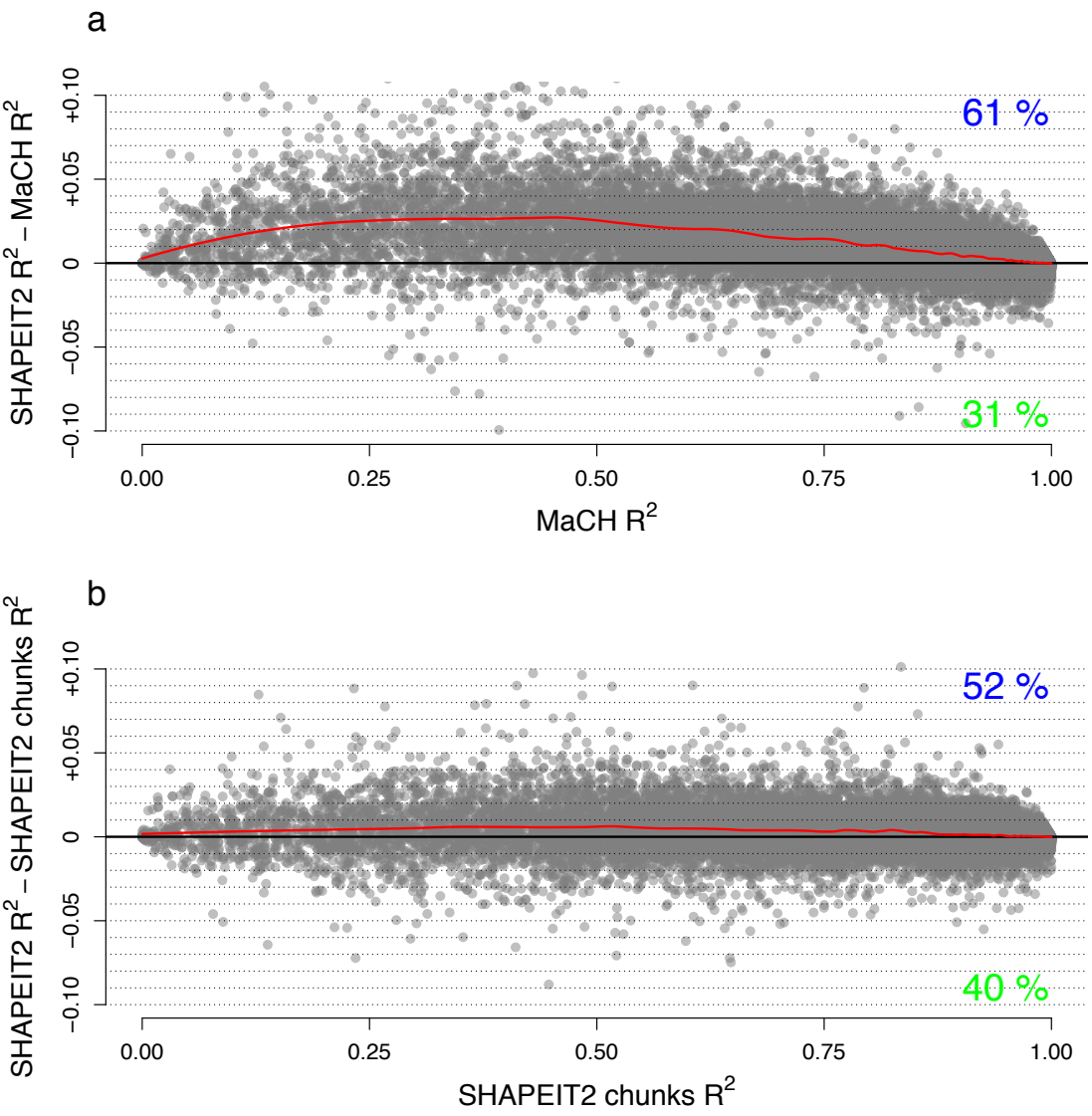
4 Supplementary Figure 3



Supplementary Figure 3 : Comparison of whole chromosome phasing methods for imputation of the WTCCC2 samples For each imputed SNP, the R^2 correlation coefficient between the true and the imputed genotypes obtained when imputing into Beagle haplotypes (panel a) and HAPI-UR haplotypes (panel b) is plotted against the R^2 difference obtained when imputing into SHAPEIT2 haplotypes. The red solid line is a curve fitted to the data points using R loess function. And the percentages in blue and green give the proportions of SNPs that have respectively a better or a worse imputation quality when using SHAPEIT2 haplotypes.

Nature Methods: doi:10.1038/nmeth.2307

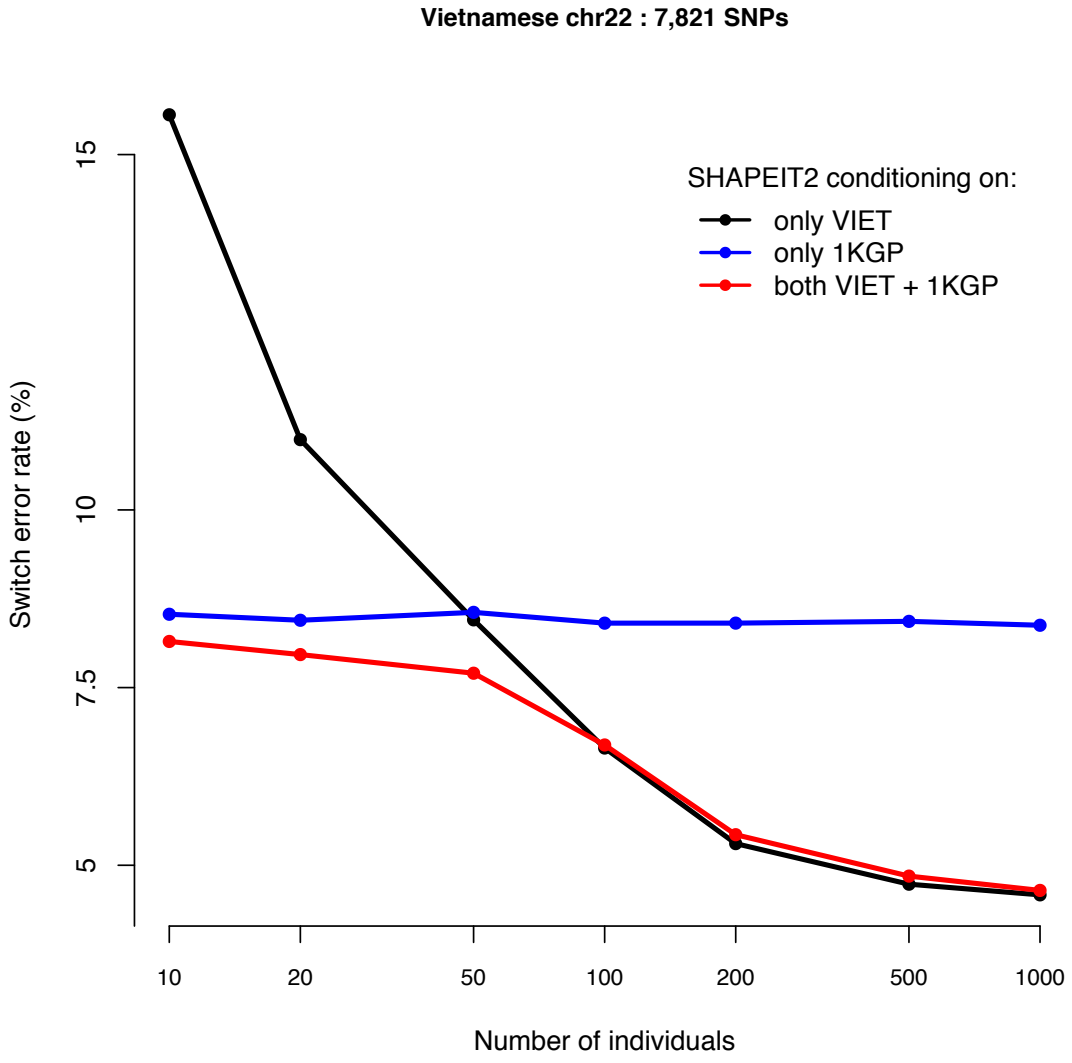
5 Supplementary Figure 4



Supplementary Figure 4 : Comparison of methods for imputation of the WTCCC2 samples. For each imputed SNP, the R^2 correlation coefficient between the true and the imputed genotypes obtained when imputing into MaCH haplotypes chunks (panel a) and SHAPEIT2 haplotypes chunks (panel b) is plotted against the R^2 difference obtained when imputing into SHAPEIT2 whole chromosome haplotypes. The red solid line is a curve fitted to the data points using R loess function. And the percentages in blue and green give the proportions of SNPs that have respectively a better or a worse imputation quality when using SHAPEIT2 haplotypes.

Nature Methods: doi:10.1038/nmeth.2307

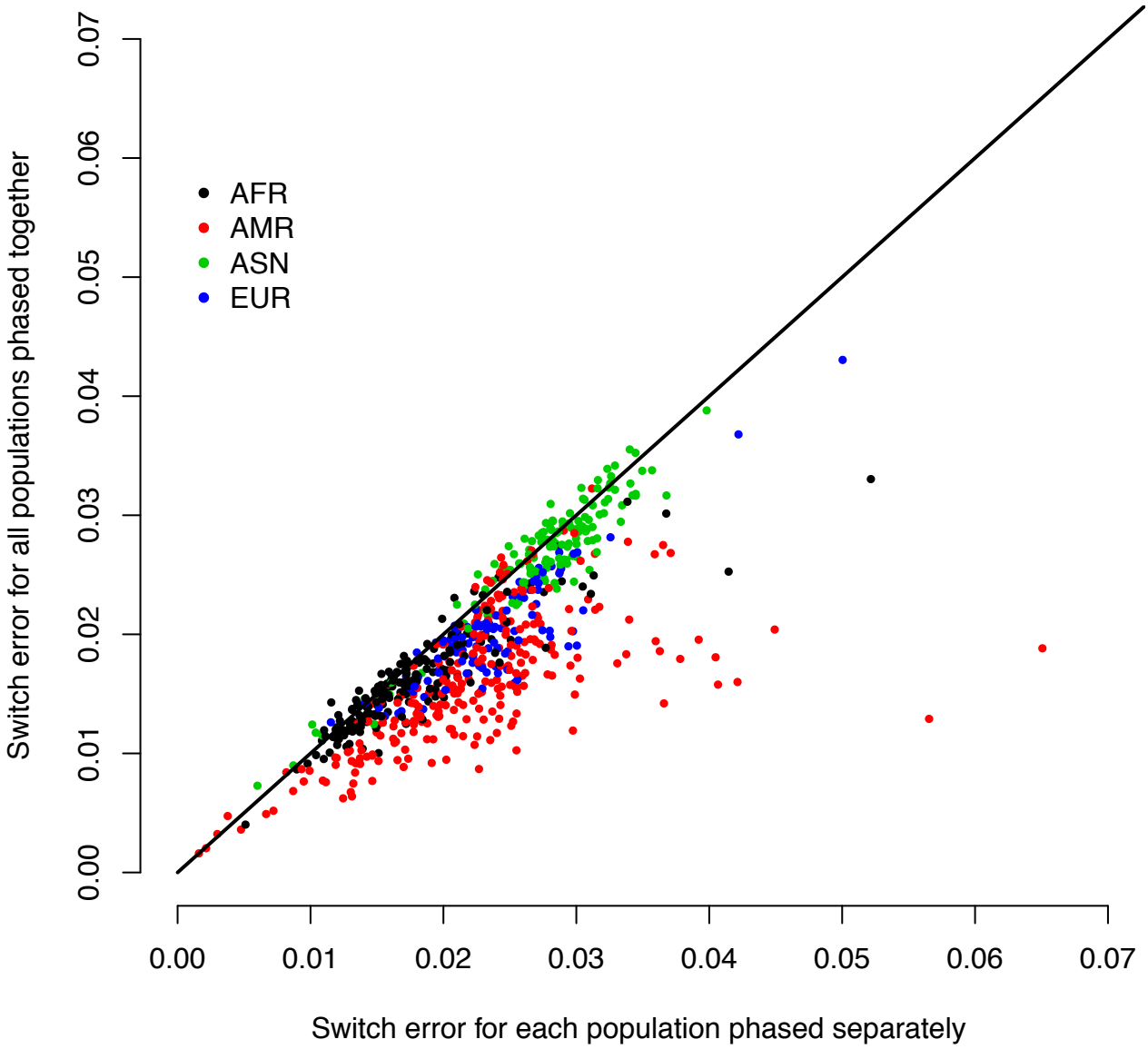
6 Supplementary Figure 5



Supplementary Figure 5 : Assessment of the benefit of using an external reference set of haplotypes Phasing performance is measured in terms of switch error percentage. The black line shows the performance of the standard SHAPEIT2 approach in which only the other Vietnamese samples in the dataset are used when constructing the conditioning set of haplotypes. The red line shows the performance of the SHAPEIT2 approach when the external set of haplotypes from the 1000 Genomes Project (1KGP) are added in to the set of conditioning haplotypes. The blue line shows the performance of SHAPEIT2 when only the 1KGP haplotypes are used as the set of conditioning haplotypes.

Nature Methods: doi:10.1038/nmeth.2307

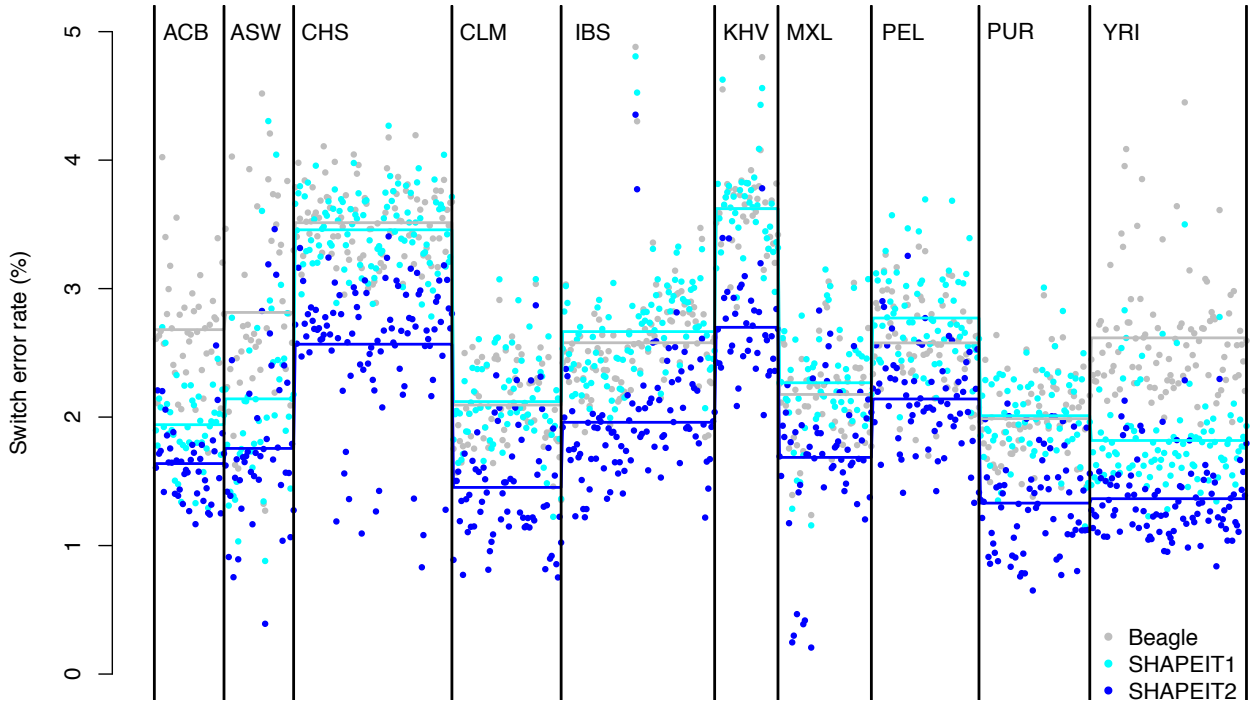
7 Supplementary Figure 6



Supplementary Figure 6 : Strategies for phasing multi-population datasets Comparison of two strategies to phase the Illumina Omni2.5 dataset. Switch error obtained when the phasing is performed independently in 4 continental groups AFR, AMR, ASN and EUR (x-axis) is plotted against the accuracy obtained when all individuals are phased together in a single group (y-axis).

Nature Methods: doi:10.1038/nmeth.2307

8 Supplementary Figure 7



Supplementary Figure 7 : Comparison of methods on multi-population datasets Methods comparison on the Illumina Omni2.5 dataset stratified per population of the 1000 Genomes project. The points give the switch error percentages obtained for the samples using SHAPEIT1 (cyan), SHAPEIT2 (blue), and Beagle (grey). Horizontal lines within each population group represent the mean switch error of that population.

Nature Methods: doi:10.1038/nmeth.2307

9 Supplementary Methods

9.1 Background

We recently described a new method for inferring haplotype phase from genotype data called SHAPEIT version 1 (SHAPEIT1) [4] which has improved accuracy and computational efficiency compared to several other methods. Inference is carried out using a Gibbs sampling approach in which the haplotypes of each individual are iteratively updated. At the core of this approach is a hidden Markov model (HMM) [8] that is used to model the conditional distributions of the Gibbs sampler, where of an individual's haplotypes are updated conditional upon the current haplotype estimates of all other samples.

This is the same general approach used by the methods Phase[15], Impute2[6] and MaCH [9]. The methods differ in the details of how they implement this approach, with all of them making some level of approximation. The methods need to make approximations because the HMM calculations involved in sampling an individual's haplotypes from the conditional distributions scale quadratically with the number of haplotypes being conditioned on. If the number of the individuals being phased is N then the complexity of the algorithm is $O(N^2)$ and this is an undesirable feature. MaCH approximates the algorithm by choosing a random set of K haplotypes to condition on at each iteration. This controls the complexity of the algorithm at $O(K^2)$. Impute2 also chooses a subset but does so in a non-random way. The method chooses the most similar subset to the previous haplotype estimates of the individual. A nice property of this approach is that the method can adapt to the haplotype structure underlying the samples being phased. For example, when phasing an admixed sample in a region where the sample has one haplotype from two distinct ancestries, the method would likely choose haplotypes from each distinct ancestry.

SHAPEIT1 introduced two new approximations. The first one involves collapsing the haplotypes being conditioned on at each iteration into a graph that captures the haplotype structure in a parsimonious way. Locally the haplotypes are clustered into K states. The HMM calculations are then carried out on this graph rather than the original set of

Nature Methods: doi:10.1038/nmeth.2307

haplotypes. The advantage of this approach is that the graph is constructed using all of the haplotypes rather than a subset. Compared to methods that choose a subset of K haplotypes this graph will capture more of the haplotype structure within each local region. The second approximation involves a way of representing the space of possible haplotypes that are consistent with an individual's genotypes. Each individual's vector of genotypes is split up into segments containing B heterozygous sites and so has just 2^B consistent haplotypes. Segments are then taken to be nodes of a graph each with 2^B states. We use $B = 3$ so that there are 8 states in each node. A path through the graph then represents a consistent haplotype across a whole chromosome. The joint distribution of all paths through the graph, conditional upon the haplotypes of other individuals, is then modelled as a HMM and transition probabilities between segments can be calculated using a forward-backward algorithm. These transition probabilities can be used to sample pairs of consistent haplotypes very quickly. A key property of this approach is that the HMM calculation involved have complexity $O(2^B K) = O(8K)$ which is typically much less than $O(N^2)$ of the full algorithm, or the $O(K^2)$ of the subset selection based methods. This allows many more haplotypes to be conditioned upon for the same computational effort. The SHAPEIT1 algorithm combines both these approximations together. In addition, a pruning and merging strategy is used in which unlikely states within each segment are carefully pruned away as the algorithm progresses. Consecutive segments are merged together when they each contain only a few likely states.

The algorithm was compared to Impute2 [6], MaCH [9], Beagle [3] and Fastphase [14] using genotyping chip data across whole chromosomes from three different datasets with 740, 925 and 1,229 samples respectively. For example, on a dataset consisting of 1,229 phase known samples at 7,985 SNPs on chromosome 22, created from a set of Vietnamese father-mother-child trios, the average distance between phasing errors (switch errors) for the methods Fastphase, Beagle, MaCH, Impute2 and SHAPEIT1 was 0.3Mb, 0.38Mb, 0.40Mb, 0.44Mb and 0.56Mb respectively. This clearly shows that the SHAPEIT1 approach offers a substantial improvement in performance over these other methods on datasets similar to those used in genome-wide association studies. This approach has recently been used in studies focussing on detecting interactions between SNPs [16].

Nature Methods: doi:10.1038/nmeth.2307

One caveat of these results is that the Impute2 results were obtained by applying the method across a whole chromosome at once. Impute2 chooses a subset of "surrogate family" conditioning haplotypes by similarity and this idea is designed to work well on relatively small regions. This idea is analogous to "surrogate parent" phasing approaches that have been suggested in situations where extended sharing of segments between individuals can be expected. For example, in more isolated populations or when a substantial fraction of a population has been sampled [7, 12]. In samples where much shorter segments are expected to be shared between individuals Impute2 searches for a subset of haplotypes (referred to as the surrogate family) that jointly capture the haplotype variation of each individual in a local region. The standard advice when using Impute2 is that regions of size around 5Mb should be used, although the approach does seem remarkably robust to the choice of region since it still outperforms MaCH, Beagle and Fastphase when run across a whole chromosome. When run on smaller 5Mb regions the Impute2 algorithm seems to be as accurate as SHAPEIT1 (see Supplementary Fig 3 in [4]). These results led us to explore the idea of combining the best features of Impute2 and SHAPEIT1 into a new algorithm. This paper describes how we have generalized the Impute2 "surrogate family" phasing approach so that it can be applied across a whole chromosome. This method has been combined with the SHAPEIT1 linear complexity method for sampling haplotypes consistent with an individual's genotypes. Adopting the Impute2 approach replaces the use of the compact HMM introduced in SHAPEIT1. Our new method is called SHAPEIT version 2 (or SHAPEIT2 for short). In the following Methods section we describe the algorithm in detail.

9.2 Notation

We assume that we are trying to estimate the haplotypes of N unrelated individuals with complete genotype data at L biallelic SNPs. The extension of the method for mother-father-child trios and parent-child duos is described later. Since the algorithm uses a Gibbs sampling scheme in which each individual's haplotypes are sampled conditional upon the current estimates of all others it is sufficient for us to consider the details of

Nature Methods: doi:10.1038/nmeth.2307

a single iteration. So we consider sampling the haplotypes of the ith individual and we call this individual's genotype vector $G = \{G_1, \ldots, G_L\}$, where $G_l \in \{0, 1, 2\}$ is the genotype at the lth SNP and represents the counts of the allele coded as 1 at the SNP. The two alleles at each SNP are abitrarily coded as 0 and 1. We use $H = \{H_1, \ldots, H_K\}$ to denote the K current haplotype estimates being used in the iteration, where $H_k = \{H_{k1}, \ldots, H_{kL}\}$ is the kth haplotype in this set with H_{kl} denoting the allele carried by this haplotype at the lth SNP. The K haplotypes are usually a subset from the total set of $2N$ haplotypes.

We want to sample a pair of haplotypes (g_1, g_2), called a diplotype, that is compatible with G so that the sum of the alleles is equal to G, denoted $G = g_1 + g_2$. Each of the haplotypes, g_1 and g_2, is a vector of alleles of length L. We divide the vector G into a number, C, of consecutive non-overlapping segments such that each segment contains B heterozyogous genotypes. We use $B = 3$. Boundaries between segments can be arbitrarily assigned when there is more than one possibility. We denote this segmentation of sites using labels $S_l \in \{1, \ldots, C\}$ that detail which segment site l resides in. We use the notation $\{s\}$ to denote the set of sites with the sth segment.

A consequence of this segmentation is that there will only be 2^B possible haplotypes that are consistent with G within each segment. These consistent haplotypes are easily enumerated and labelled from 1 to 2^B. For example, Figure S1 shows the four haplotypes that are consistent with a segment of G that is 01012. Based on this construction we can represent a haplotype that is consistent with G across its whole length as a vector of labels $X = \{x_1, \ldots, x_L\}$ where x_l denotes the label of the consistent haplotype at the lth site in the S_lth segment, with the restriction that the labels are identical within each segment.

9.3 A segmented haplotype model

A given realization of X precisely determines a haplotype consistent with G, however X is unobserved. To estimate X we need to build a model for the conditional distribution of X given H, $P(X|H)$. Sampling X from this distribution will then form the iteration

Nature Methods: doi:10.1038/nmeth.2307

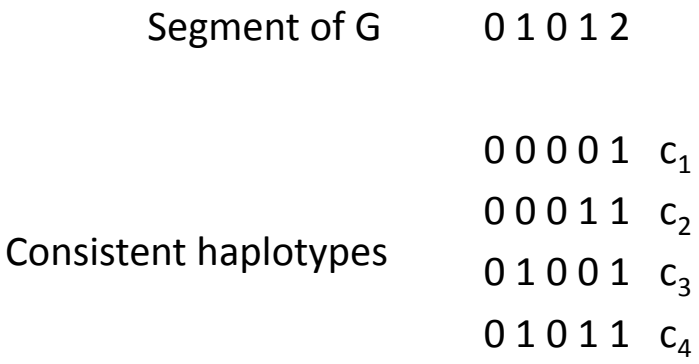


Fig S1 The figure shows an example of a segment of a genotype vector G that has $B = 2$ heterozygous sites. Also shown are the $2^B = 4$ haplotypes that are consistent with the segment of G. The haplotypes are labelled c_1, c_2, c_3, c_4. The haplotypes occur in consistent pairs so that $G = c_i + c_{2^B+1-i}$ where $i \in \{1, \ldots, 2^{B-1}\}$. In this example the pairs of consistent haplotypes are (c_1, c_4) and (c_2, c_3).

Nature Methods: doi:10.1038/nmeth.2307

of the Gibbs sampler that we require.

We assume a haplotype consistent with G can be modelled as an imperfect mosaic of the haplotypes in H. We model this using the same HMM model underlying Phase[15], Impute2[6] and MaCH[9]. In this model the sequence of unobserved states $Z = \{Z_1, \ldots, Z_L\}$ are such that $Z_l \in \{1, \ldots, L\}$ denotes the haplotype being 'copied' at site l. The transition probabilities between the states along the sequence are governed by an estimate of the fine-scale recombination rate between adjacent sites as follows

$$\begin{aligned} Pr(Z_1 = u) &= \frac{1}{K}, \\ P(Z_l = u | Z_{l-1} = v) &= \begin{cases} e^{-\frac{\rho_l}{K}} + \frac{1-e^{-\frac{\rho_l}{K}}}{K} & u = v, \\ \frac{1-e^{-\frac{\rho_l}{K}}}{K} & u \neq v, \end{cases} \end{aligned}$$

where $\rho_l = 4N_e r_l$ and r_l is the per generation genetic distance between sites l and $l-1$ and N_e is the effective population size. We use $N_e = 15,000$. On the human datasets we have analysed the results are very insensitive to changes in this parameter.

We now have two unobserved sequences X and Z and we are primarily interested in making inference about X. We do this by constructing the joint conditional distribution of X and Z given H as

$$P(X, Z|H) = p(X_1|Z_1, H)P(Z_1)\prod_{l=2}^{L} p(X_l, Z_l|X_{l-1}, Z_{l-1}, H).$$

The joint transition probabilities are constrained by the restriction described above that labels of consistent haplotypes at consecutive sites within a segment must be identical. Consequently, when sites l and $l-1$ are within the same segment i.e. $S_l = S_{l-1}$, we have

$$p(X_l = i, Z_l = u | X_{l-1} = j, Z_{l-1} = v, H) = \begin{cases} P(X_l = i|Z_l = u, H)P(Z_l = u|Z_{l-1} = v) & i = j \\ 0 & \text{otherwise,} \end{cases}$$

and when sites l and $l-1$ are in different segments i.e. $S_l \neq S_{l-1}$, we have

$$p(X_l = i, Z_l = u | X_{l-1} = j, Z_{l-1} = v, H) = P(X_l = i|Z_l = u, H)P(Z_l = u|Z_{l-1} = v).$$

The emission probabilities of the model are given by

$$P(X_l = i|Z_l = u, H) = \begin{cases} \lambda, & H_{ul} \neq A_{li} \\ 1-\lambda, & H_{ul} = A_{li} \end{cases}$$

Nature Methods: doi:10.1038/nmeth.2307

where we use A_{lb} to denote the allele carried by the bth consistent haplotype at site l. Also, $\lambda = \frac{\theta}{2(\theta+K)}$ and $\theta = \left(\sum_{i=1}^{K-1} \frac{1}{i}\right)^{-1}$ [8].

9.4 Sampling consistent haplotypes

We use the model in the previous subsection to calculate marginal probabilities $P(X_1 = i|H)$ and transition probabilities between the labels of consistent haplotypes at consecutive sites, $P(X_l = i|X_{l-1} = j, H)$. The precise details of all the calculation involved are provided in the following subsection. It is important to highlight that these calculations are linear in the number of haplotypes being conditioned on, K, because both the state spaces of the X and Z are representing haplotypes rather than diplotypes.

We then use these marginal and transition probabilities to sample consistent haplotypes for each individual. Since each individual has two unobserved haplotypes we need to sample two consistent haplotypes at the same time. We do this by first sampling vectors of consistent haplotype labels across all sites and then converting this vectors of labels into realized haplotyes. We use notation $X_l^{(1)}$ and $X_l^{(2)}$ to denote the pair of vectors of consistent haplotype labels that we sample.

In each segment there will be 2^B consistent haplotypes and these will occur in 2^{B-1} consistent pairs. The scheme we use to sample consistent haplotypes is described by the following steps

1. A pair of consistent haplotypes in the first segment with labels (i, j) is sampled with probability proportional to $P(X_1 = i|H)P(X_1 = j|H)$. Not all pairs of labels will denote pairs of consistent haplotypes. In the example in Figure S1 above we would only consider label pairs (c_1, c_4) and (c_2, c_3). It is also worth noting here that the sampler uses the marginal probabilities of the first *site* X_1 but this is sufficient since all sites in the same segment will have identical marginal probabilities due to the constraints placed on the sequence X. We set the labels of all sites in the first segment to be the same as those sampled for the first site. That is,

$$X_l^{(1)} = i, X_l^{(2)} = j \quad \forall \quad l \in \{1\}$$

Nature Methods: doi:10.1038/nmeth.2307

2. Set r = 2.

3. Let $q(r)$ be the index of the first site in the rth segment. We consider the rth segment and use the transition probabilities between the last site in segment $r-1$ and the first site in segment r, $P(X_{q(r)}|X_{q(r)-1}, H)$ say. A pair of consistent haplotypes in the second segment with labels (d, f) is sampled with probability proportional to $P(X_{q(r)} = d|X_{q(r)-1} = i, H)P(X_{q(r)} = f|X_{q(r)-1} = j, H)$. We set the labels of all sites in the rth segment to be the same as those sampled for site $q(r)$. That is,

$$X_l^{(1)} = d, X_l^{(2)} = f \quad \forall \quad l \in \{r\}$$

4. Set $r = r + 1$.

5. If $r = C + 1$ then stop, else go to Step 3. In other words, process all C segments in the same way.

The result is a pair of vectors of consistent haplotype labels, $X^{(1)}$ and $X^{(2)}$, across the whole region being phased and these can be turned into new haplotype estimates, (g_1, g_2), using $g_{il} = A_{lX_l^{(i)}}$ for $i \in \{1, 2\}$ and $l \in \{1, \ldots, L\}$. These haplotype estimates can then be added back into the haplotype set H and the next individual's haplotypes can be estimated, although their current haplotype estimates must be removed from H first.

SHAPEIT2 can also handle mother-father-child trios and parent-child duos. These situations impose additional constraints on the configurations of consistent haplotypes within each segment. When processing a trio the haplotypes of the mother, father and child are updated jointly. First, a set of segments are defined so that each one contains no more than 16 possible haplotypes across all members of the family. Only sites that are heterozygous in all members of the nuclear family are phase-unknown and it is such sites that increase that number of possible haplotypes. Then, haplotypes in each segment are labelled as paternal or maternal, and transmitted or untransmitted, with the childs haplotypes corresponding to the pair of transmitted haplotypes. Finally sets of 4 haplotypes are sampled across the segments such that they conform to these Mendel rules and solve all the members of the family. Parent-child duos are handled in an analogous way. It

Nature Methods: doi:10.1038/nmeth.2307

is worth pointing out that this method of handling closely related samples is very computationally efficient. The constraints imposed by the family structure mean that many sites are phase-known so that sites with phase ambiguity will be more sparsely spread out across a chromosome than for unrelated samples. This means that the segments span larger regions and there are fewer segments across a chromosome and the algorithm takes less time to process.

SHAPEIT2 can also handle sporadic missing genotypes in the dataset. Missing data in an individual increases the number of combinations of consistent haplotypes that an individual could carry. For example, if the genotypes in region consist of a single missing genotype and a single heterozygous site then there will be four consistent haplotypes in this region and four combinations of these consistent haplotypes that the individual could carry. In contrast, if the region contained just two heterozygous sites then there will also be four consistent haplotypes in this region but only two combinations of these consistent haplotypes that the individual could carry.

We have also added options to the program that allow reference sets of haplotypes to be used when phasing a set of unphased samples. This set of haplotypes are added to the set of the current haplotype estimates at each iteration so that they can be used as conditioning haplotypes when updating each individual's haplotypes.

9.4.1 Algorithmic details

In this section we set out the precise details of the calculations of the probabilities $P(X_1 = i|H)$ and $P(X_l = i|X_{l-1} = j, H)$ that we need to sample consistent haplotypes. To do this we use the notation $G_{n,m}$ to denote the model of the data from site n to m and define the partial vectors $X_{n,m} = \{X_n, \ldots, X_m\}$, $Z_{n,m} = \{Z_n, \ldots, Z_m\}$. We then define forward probabilities as

$$a_m(i, u) = P(X_m = i, Z_m = u, G_{1,m}|H).$$

These are initialized as

$$a_1(i, u) = P(X_1 = i|Z_1 = u, H)P(Z_1 = u).$$

Nature Methods: doi:10.1038/nmeth.2307

Subsequent probabilities are defined recursively, but the recursions are different dependent upon whether the mth and $(m+1)$th sites are in the same segment. So when $S_m = S_{m+1}$ we have

$$\begin{aligned} a_{m+1}(i,u) &= P(X_{m+1}=i|Z_{m+1}=u,H)\sum_{v=1}^{K} a_m(i,v)P(Z_{m+1}=u|Z_m=v) \\ &= P(X_{m+1}=i|Z_{m+1}=u,H)\Big[\frac{1-e^{-\frac{\rho_l}{K}}}{K}a_m(i,\bullet)+e^{-\frac{\rho_l}{K}}a_m(i,u)\Big], \end{aligned}$$

and when $S_m \neq S_{m+1}$ we have

$$\begin{aligned} a_{m+1}(i,u) &= P(X_{m+1}=i|Z_{m+1}=u,H)\sum_{j=1}^{2^B}\sum_{v=1}^{K} a_m(j,v)P(Z_{m+1}=u|Z_m=v) \\ &= P(X_{m+1}=i|Z_{m+1}=u,H)\Big[\frac{1-e^{-\frac{\rho_l}{K}}}{K}a_m+e^{-\frac{\rho_l}{K}}a_m(\bullet,v)\Big] \end{aligned}$$

where $a_m(i,\bullet)=\sum_{u=1}^{K} a_m(i,u)$, $a_m(\bullet,u)=\sum_{i=1}^{2^B} a_m(i,u)$ and $a_m=\sum_{i=1}^{2^B}\sum_{u=1}^{K} a_m(i,u)$. The calculation of the quantities $a_m(i,\bullet)$, $a_m(\bullet,u)$ and a_m saves a significant amount of computation.

Similarly, we define backward probabilities as

$$b_m(i,u) = P(G_{m+1,M}|X_m=i,Z_m=u,H).$$

These are initialised as

$$b_M(i,u)=1.$$

Subsequent probabilities are also defined recursively, but the recursions are different dependent upon whether the mth and $(m+1)$th sites are in the same segment. So when $S_m = S_{m+1}$ we have

$$\begin{aligned} b_m(i,u) &= \sum_{v=1}^{K} b_{m+1}(i,v)P(X_{m+1}=i|Z_{m+1}=v)P(Z_{m+1}=v|Z_m=u) \\ &= \Big[\frac{1-e^{-\frac{\rho_l}{K}}}{K}B_{m+1}(i,\bullet)+e^{-\frac{\rho_l}{K}}b_{m+1}(i,u)P(X_{m+1}=i|Z_{m+1}=u)\Big] \end{aligned}$$

and when $S_m \neq S_{m+1}$ we have

$$\begin{aligned} b_m(i,u) &= \sum_{i=1}^{2^B}\sum_{v=1}^{K} b_{m+1}(i,v)P(X_{m+1}=i|Z_{m+1}=v)P(Z_{m+1}=v|Z_m=u) \\ &= \Big[\frac{1-e^{-\frac{\rho_l}{K}}}{K}B_{m+1}+e^{-\frac{\rho_l}{K}}B_{m+1}(\bullet,u)\Big] \end{aligned}$$

Nature Methods: doi:10.1038/nmeth.2307

where $B_m(i,\bullet) = \sum_{u=1}^{K} b_m(i,u)P(X_m = i|Z_m = u)$, $B_m(\bullet,u) = \sum_{i=1}^{2^B} b_m(i,u)P(X_m = i|Z_m = u)$ and $B_m = \sum_{i=1}^{2^B}\sum_{u=1}^{K} b_m(i,u)P(X_m = i|Z_m = u)$.

The forward and backwards probabilities can then be used to calculate the marginal probability of the unobserved states, X_m and Z_m, at site m as

$$\begin{aligned} P(X_m = i, Z_m = u|H) &\propto P(X_m = i, Z_m = u, G_{1,m}|H)P(G_{m+1,M}|X_m = i, Z_m = u, H) \\ &= a_m(i,u)b_m(i,u). \end{aligned}$$

The marginal probability of the unobserved state, X_m, is then obtained as

$$P(X_m = i|H) \propto \sum_{u=1}^{K} a_m(i,u)b_m(i,u),$$

and this allows us to calculate the marginal probability distribution of the first site, $P(X_1 = i|H)$.

Along similar lines we can calculate the *joint* marginal probability of the unobserved states at sites m and $m+1$ as

$$\begin{aligned} P(X_m = i_1, Z_m = u_1, X_{m+1} = i_2, Z_{m+1} = u_2|H) &\propto a_m(i_1,u_1)P(Z_{m+1} = u_2|Z_m = u_1) \\ &\quad \times P(X_{m+1} = i_2|Z_{m+1} = u_2)b_{m+1}(i_2,u_2). \end{aligned}$$

This can then be used to calculate the transition probabilities we need

$$P(X_{m+1} = i_2|X_m = i_1, H) = \frac{\sum_{u_1=1}^{K}\sum_{u_2=1}^{K} P(X_m = i_1, Z_m = u_1, X_{m+1} = i_2, Z_{m+1} = u_2|H)}{\sum_{i_2=1}^{2^B}\sum_{u_1=1}^{K}\sum_{u_2=1}^{K} P(X_m = i_1, Z_m = u_1, X_{m+1} = i_2, Z_{m+1} = u_2|H)}.$$

Calculation of the forward and backward probabilities can sometimes result in very small probabilities that go below the lower limit for a floating point datatype. A standard solution to this problem [13] involves rescaling the probabilities by an arbitrary constant when probabilities become small and we implement this approach when needed.

9.5 Surrogate family phasing

Experience suggests that the "surrogate family" phasing idea in Impute2 whereby the K haplotypes being conditioned upon at each iteration are chosen to be the closest K

Nature Methods: doi:10.1038/nmeth.2307

haplotypes to the current haplotypes works well over reasonably short regions around 5Mb long, but not as well across whole chromosomes. Our new method selects subsets of haplotypes in local regions but carries out inference across whole chromosomes at once. Within the framework of the SHAPEIT method we apply the model in overlapping chunks with the constraint that the pairs of consistent haplotypes must agree in the overlapping regions between the chunks. As above, these steps relate to the Gibbs sampling update of a single individual conditional upon the current haplotype estimates of all other individuals. The method is implemented using the following steps.

1. The whole chromosome is divided up into consecutive non-overlapping *segments* of B heterozygous sites. The boundaries of these segments stay constant through the first two stages of iterations. The segments are then redefined after the pruning and merging process. This is described in more detail in the next section.

2. The whole chromosome is divided up in a second way into overlapping *windows* of length W Mb. Window size can be specified by the user. The windows are generally much larger than the segments with each window containing many segments. The overlapping regions between these *windows* consist of one *segment*, as defined by step 1. The windows are determined using a stochastic algorithm that recursively partitions the chromosome and results in a set of windows that are typically around WMb in length. The windows are re-defined on each iteration so that the overlapping segment between windows changes at each iteration. This reduces any bias that might occur if a fixed set of windows was chosen.

3. Within the first window a subset of K haplotypes are chosen from H that are informative for the phasing of G within the window. The haplotypes in H are scored according to how similar they are to the current haplotype estimates for the individual being updated. The K haplotypes with the lowest scores are selected. When developing this algorithm we found it necessary to adjust this method to allow for windows where the individual being updated shares two alleles identically by de-

Nature Methods: doi:10.1038/nmeth.2307

scent with another individual at a significant proportion of sites in the window. This can happen due to cryptic relatedness between samples. In the following two sub-sections we give details of a basic method of haplotype selection (Algorithm 1) and an adjusted version (Algorithm 2) that is robust to cryptic relatedness and is implemented in SHAPEIT2. The model is applied to all the segments in the first window conditional upon the subset of K haplotypes. Then a pair of consistent haplotypes is sampled across the window.

4. The model is then applied to the second window using a new set of K haplotypes selected for that window. The sampling of consistent haplotypes across this window is initialized in it's first segment using the pair of consistent haplotypes in the last segment of the first chunk and then the calculated transition probabilities are used to sample further segments.

5. The method deals with all subsequent windows in the same way as the second window until a complete sequence of consistent haplotypes across the whole chromosome is sampled.

9.5.1 Algorithm 1

We rank each haplotype h in H by the distance between h and the current haplotype guess $G = (g_1, g_2)$, defined as

$$D(h) = \min[\texttt{hamming}(h, g_1), \texttt{hamming}(h, g_2)],$$

where $\texttt{hamming}(h, g)$ is the Hamming distance between haplotypes h and g i.e. the number of alleles that differ between the two haplotypes g and h. If $\texttt{hamming}(h, g)$ is small, that means that h is similar to g and is likely to be useful for updating the haplotypes of G. Conversely, if $\texttt{hamming}(h, g)$ is large, that means that h differs a lot from g, which suggests that h may not be very informative for updating G. Thus, if we use only the

Nature Methods: doi:10.1038/nmeth.2307

haplotypes of H that have the smallest scores, we should be able to base the estimation only on the most informative part of H. Note that the computation time required for the calculation of this score for all the haplotypes in H is negligible compared to the calculations involved in Appendix A. Pseudo code for the algorithm is given below.

Algorithm 1 Selection of the K best haplotypes

for all $h \in H$ **do**

Calculate $D(h)$.

end for

Sort H by increasing $D(h)$ values.

return K first haplotypes in H.

When developing this approach we found instances when Algorithm 1 did not perform well where two individuals shared *both* alleles identically by descent (IBD) across a large proportion of a given window. This can happen if the two individuals are closely related. For example, if two sibs are included in the sample set then they will tend to share large segments of the genome with an IBD count of two. We observed that use of Algorithm 1 resulted in high switch error rates for such pairs of samples in such regions. We tracked this down to a convergence problem where each individual's haplotypes are updated to be close to identical to the haplotypes of the other individual at each iteration, which tends to fix switch errors in these samples, and the algorithm can not move away from this solution. To overcome this problem, we developed a refined the haplotype subset strategy in order to be more robust to this case.

9.5.2 Algorithm 2

The main idea of this algorithm is to identify cases where we think that a given individual, I, shares a large proportion of the window with an IBD count of two with the individual being updated, G, and then make it less likely that the haplotypes of I are included in the subset of K haplotypes. To do this we consider the two haplotypes of each individual in turn $I = (i_1, i_2)$ and compare them to the two haplotypes of $G = (g_1, g_2)$. We look for cases where there is close *matching* between distinct pairs of haplotypes in I and G.

Nature Methods: doi:10.1038/nmeth.2307

More precisely, we check for whether either of the following two conditions occur.

9.5.3 Matching conditions

1. $\texttt{hamming}(i_1, g_1) < \texttt{hamming}(i_1, g_2)$ and $\texttt{hamming}(i_2, g_2) < \texttt{hamming}(i_2, g_1)$
2. $\texttt{hamming}(i_1, g_2) < \texttt{hamming}(i_1, g_1)$ and $\texttt{hamming}(i_2, g_1) < \texttt{hamming}(i_2, g_2)$

Having identified these pairs of haplotypes we then look at the pattern of IBD sharing across the window. We use g_1^* and g_2^* to denote the haplotypes of G that match i_1 and i_2 respectively. So for condition 1 we would have $g_1^* = g_1$ and $g_2^* = g_2$ and for condition 2 we would have $g_1^* = g_2$ and $g_2^* = g_1$ respectively. We then look to see where i_1 matches g_1^*, where i_2 matches g_2^* and then at the overlap between this matching. This can be summarised by the following three numbers

$$\begin{aligned} S(i_1) &= \text{number of sites where } i_1 \text{ matches } g_1^* \\ S(i_2) &= \text{number of sites where } i_2 \text{ matches } g_2^* \\ S(i_1, i_2) &= \text{number of sites where } i_1 \text{ matches } g_1^* \text{ AND } i_2 \text{ matches } g_2^* \end{aligned}$$

Given these measures we can define the proportion, p_1, of sharing between i_1 and g_1^* that is also shared between i_2 and g_2^*

$$p_1 = \frac{S(i_1, i_2)}{S(i_1)} \tag{1}$$

And similarly, the proportion, p_2, of sharing between i_2 and g_2^* that is also shared between i_1 and g_1^*

$$p_2 = \frac{S(i_1, i_2)}{S(i_2)} \tag{2}$$

This results in two measures p_1 and p_2 that are close to 1 when the overlap between matches of i_1 with g_1^* and matches of i_2 with g_2^* is large. This is the situation we wish to avoid. Conversely, they are close to 0 when this overlap is small. This situation is less worrying for our algorithm as it means there less chance that there is a shared segment of IBD=2 within the window. Our algorithm then chooses to exclude i_1 from consideration for the K subset with probability p_1 and similarly to exclude i_2 from consideration for the K subset with probability p_2.

Nature Methods: doi:10.1038/nmeth.2307

One caveat is that the proportions p_1 and p_2 depend a lot on the frequency spectrum of the sites in the sense that if many sites are rare (for example when working with data from sequencing studies), many sites will be homozygous for the reference allele, which tends to inflate p_1 and p_2. To be more independent of the frequency spectrum of sites in the sample, we calculate $S(i_1)$, $S(i_2)$ and $S(i_1, i_2)$ only for sites where I and G are not homozygous for the same allele.

The full algorithm in pseudo code is

Algorithm 2 Selection of the K best haplotypes robust to close relatives

Create an empty set of haplotypes $W = \emptyset$.
for all $I \neq G$ **do**
  if Matching conditions 1 or 2 are satisfied **then**
    Define the matching haplotypes g_1^* and g_2^*.
    Calculate p_1 and p_2.
    Add i_1 to W with probability $1 - p_1$.
    Add i_2 to W with probability $1 - p_2$.
  end if
end for
For each haplotype in W calculate $D(h)$ and sort W by increasing $D(h)$ values.
return K first haplotypes in W.

Nature Methods: doi:10.1038/nmeth.2307

9.6 State pruning and segment merging

An iteration of our algorithm consists of updating all the individuals in a random order by using the update step described in previous sections. The algorithm starts from random haplotype estimates for each individual and performs three stages of iterations:

1. A first stage of *burn-in* iterations are run to find a better starting point. For the analysis in this paper we have used 7 burn-in iterations. The only information that is retained from these iterations are the haplotype estimates from the last iteration. These are used as the initial estimates of the second stage of iterations.

2. A second stage of *pruning* iterations are then run in which the joint probabilities, $P(X_l = i, X_{l-1} = j, H)$, for each individual are stored. For the analysis in this paper we have used 8 pruning iterations. These *haploid* level probabilities are then converted very easily into *diploid* level probabilities and these are then averaged across the iterations. These joint probabilities then represent the set of likely paths through the space of consistent haplotypes at this stage of the algorithm. We then prune away sets of states that have low probability and try to merge together segments when only very few combinations of states across segments are favoured. This defines a new (smaller) set of segments and a new (smaller) set of consistent haplotypes within each segment and overall a more parsimonious representation of the space of consistent haplotypes. This has the dual effect of focussing attention of subsequent iterations on the most likely set of haplotypes and increases computational efficiency since the model space is reduced. Full details of the state pruning and segment merging are set out in the following sub-section.

3. A third of stage with a larger number of *main* iterations are used to obtain a final estimate of the transition probabilities. These *haploid* level probabilities are then converted into *diploid* level probabilities and these are then averaged across the main iterations. These conversion and averaging is the same as that used in

   The haplotype estimate for each individual is then obtained by finding the most likely sequence of haplotypes across the segments via a Viterbi algorithm. For the

Nature Methods: doi:10.1038/nmeth.2307

analysis in this paper we have used 20 main iterations.

9.6.1 Algorithmic details

Both the pruning and main iterations involve conversion of the haploid level state probabilities to diploid level probabilities and subsequent averaging of these diploid level probabilities. The details of these two steps are as follows

- **Conversion to diploid probabilities**. We convert the haploid state probabilities to diploid state probabilities by considering all possible pairs of consistent haplotypes. That is we consider pairs of consistent haplotypes $X^{(1)}$ and $X^{(2)}$ at sites $l-1$ and l with probabilities

$$\begin{aligned} P(X_{l-1}^{(1)} = i_1, X_{l-1}^{(2)} = i_2, X_l^{(1)} = j_1, X_l^{(2)} = j_2|H) &= P(X_{l-1}^{(1)} = i_1, X_l^{(1)} = j_1|H) \\ &\quad \times P(X_{l-1}^{(2)} = i_2, X_l^{(2)} = j_2|H) \quad (3) \end{aligned}$$

  such that $A_{(l-1)i_1} + A_{(l-1)i_2} = G_{l-1}$ and $A_{lj_1} + A_{lj_2} = G_l$. If there are $B = 3$ heterozygous sites in each segment then there will be $2^B = 8$ consistent haplotypes in each segment and within each segment there will be $2^{B-1} = 4$ pairs of consistent haplotypes. Each pair of consistent haplotypes in the segment that contains site $l-1$ can be paired with each pair of consistent haplotypes in the consecutive segment containing site l in two ways so that there will be 32 possible joint states at sites $l-1$ and l at the diploid level.

- **Averaging step.** During the pruning and main iterations we store the diploid state probabilities between segments as described by Eq.3. So for nth iteration we store the probabilities $P_n(X_{l-1}^{(1)} = i_1, X_{l-1}^{(2)} = i_2, X_l^{(1)} = j_1, X_l^{(2)} = j_2|H)$ for all possible quadruplet (j_1, j_2, i_1, i_2) of labels. We then average these joint probabilities to obtain

$$P_{avg}(X_{l-1}^{(1)} = i_1, X_{l-1}^{(2)} = i_2, X_l^{(1)} = j_1, X_l^{(2)} = j_2|H) = \frac{1}{V}\sum_{n=1}^{V} P_n(X_{l-1}^{(1)} = i_1, X_{l-1}^{(2)} = i_2, X_l^{(1)} = j_1, X_l^{(2)} = j_2|H)$$

  It is sufficient to consider only transitions between sites $l-1$ and l that straddle the boundaries of segments because within each segment no changes in state are

Nature Methods: doi:10.1038/nmeth.2307

permitted.

During the second stage of pruning iterations we use the averaged diploid probabilities to obtain a more parsimonious representation of the space of haplotypes underlying a genotype. The precised details of this process are given in the following two steps.

- **Pruning step.** We calculate which states account for the 99% of the total probability. To do this we order the probabilities in decreasing order and then discard all states after the nth state if the cumulative sum of the first nth state probabilities are greater than 99%. For example, we would call states (i_1, i_2) and (j_1, j_2) "connected" if $P_{avg}(X_{l-1}^{(1)} = i_1, X_{l-1}^{(2)} = i_2, X_l^{(1)} = j_1, X_l^{(2)} = j_2|H)$ is within the top 99% of cumulative probabilities of all such possible states. All other states pairs across segment boundaries are labelled "unconnected". We carry this out across all segment boundaries and then we remove diploid states in each segment that are "unconnected" to any diploid states in the two flanking segments. It may be the a diploid state within a segment is unconnected to any state in one the flanking segments but does have connections to diploid states in the other flanking segment. In this case, we retain the state and add in the connection to the diploid state that has the highest probability.

- **Merging step.** We consider each pair of consecutive segments and count the number of remaining diploid states. If this number is less than or equal to 8 then we merge the segments. That is we use the remaining diploid states to calculate a new set of consistent haplotypes that span both segments.

This process defines a new set of segments and a new set of consistent haplotypes within each segment. This new "model space" is used for all subsequent iterations.

9.7 Multi-threading and recommendations for whole genome phasing

For increased computational efficiency both the SHAPEIT1 and SHAPEIT2 algorithms have been implemented using multi-threading via POSIX Threads. The user can specify

Nature Methods: doi:10.1038/nmeth.2307

the number of threads to be used on the command line (the default is 1). This feature is especially useful when multiple CPUs with multiple cores are available on a compute server as it allows these resources to be used in full. For example, a machine with two quad-core CPUs would allow 8 threads to be used and each job would finish around 8 times faster. We have frequently taken advantage of servers with two CPUs each with 12 cores to run phasing jobs using upto 24 threads. The phasing of the Illumina Omni2.5 genotype data produced by the 1000 Genomes Project was carried out in this way.

When phasing a whole genome it makes sense to provide more compute power for larger chromosomes. For example, on a server with 60 cores we would suggest distributing compute resources according to chromosomal size i.e. chromosomes 1 and 2 using 5 cores each, chromosomes 3-6 using 4 cores each, chromosomes 7-12 using 3 cores each, chromosomes 13-18 using 2 cores each and chromosomes 19-22 using 1 core each. In this way each of the 22 compute jobs needed should take roughly the same amount of time to run.

Each single thread is used to process a single individual, trio or duo at a time. This involves calculating the transition probabilities between segments for the set of consistent haplotypes and then sampling a new set of consistent haplotypes. When multiple threads are used this means that multiple individuals, trios or duos can be updated at once. This does mean that the updates we use in this case are not strictly Gibbs sampling updates where each sample is updated conditionally upon the current haplotype updates of all other samples. We have never found this to be a problem on the datasets we have analysed and the results in this paper confirm this.

9.8 Metrics for comparing methods

All computational phasing methods will make errors in the estimation of haplotypes and the probability of making at least one error will increase with the length of the sequence being analysed. Across a whole chromosomes this probability will be 1 in all but the most trivial examples where samples are highly inbred. For this reason many previous phasing comparisons have used the switch error metric (SER - switch error rate) [10, 11] which

Nature Methods: doi:10.1038/nmeth.2307

measures the proportion of changes (or switches) that need to be applied to the estimated haplotypes in order to make them agree with the truth. This metric is independent of the length of sequence being analysed. In our experiments we also calculated the locations of switches for each estimated set of haplotypes and report the mean distance between switches (MSD - mean switch distance). This metric gives a clear sense of the average length of correct haplotype segments that each method will produce and can be more interpretable than switch error rate. We also recorded the running times taken by each of the methods.

Nature Methods: doi:10.1038/nmeth.2307

10 Supplementary Note 1 - Datasets

10.1 European X Chromosome dataset

This dataset consisted of 3,481 individuals with European ancestry from the cohorts GRIV 300K (355 samples), ACS 300K (411 samples), DESIR 300K (697 samples), Illumina Control 1M (49 samples), Illumina Control 550K (1,579 samples) and Illumina Control 300K (370 samples) (downloaded from www.illumina.com). On these cohorts, we (1) extracted only the SNPs included on the Illumina 300K chip, (2) extracted the X chromosomes, (3) removed SNPs with missing data rate above 0.05 and with MAF under 0.01, (4) removed individuals with missing data rate above 0.2, (5) aligned alleles to be relative to the + strand of the human reference genome, (6) imputed sexes according to the level of heterozygous SNPs, (7) removed females and finally (8) set all remaining heterozygous SNPs as missing. This resulted in 1,850 male phase known haplotypes at 7,821 SNPs. These haplotypes were then randomly paired to produce 925 phase known samples.

10.2 WTCCC2 X Chromosome dataset

We used 1,480 male samples from the WTCCC2 control dataset. These samples were genotyped on the Affy 6.0 genotyping chip. We used the merged and filtered dataset released by the project. The male haplotypes were randomly paired to produce 740 phase known samples at 29,227 SNPs.

10.3 Vietnamese chromosome 22 dataset

597 trios and 35 duos were extracted from a cohort of Vietnamese families who were genotyped in the context of leprosy with Illumina 660K human beadchips. This cohort is an extended version of previously described studies [1, 2]. The various software tested were used to phase the 1,229 parents at 7,985 SNPs that passed QC filtering on chromosome 22. Respective switch error rates were measured between heterozygous sites for which

Nature Methods: doi:10.1038/nmeth.2307

the phase was known through the Mendel transmission rules (i.e. sites that are not triple heterozygous for trios and not double heterozygous for duos). When we combined this dataset with the 1000 Genomes Phase I haplotypes as an external reference we removed 64 SNPs as these were not present in the 1000 Genomes dataset or had inconsistencies in the alleles reported.

10.4 1000 Genomes Illumina Omni2.5 dataset

We obtained the chromosome 20 estimated haplotypes from the 1000 Genomes samples genotype using the Illumina Omni2.5 chip from the ftp site `ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/working/20111117_omni_genotypes_and_intensities/`. A detailed summary of this dataset is given below in Table S1.

10.5 1000 Genomes Phase I haplotypes

We obtained the chromosome 20 estimated haplotypes from the 1000 Genomes samples sequenced at low-coverage ($\sim$ 4X) from the ftp site `ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/release/20110521/`

10.6 High-coverage sequencing data on a European mother-father-child trio

We used a CEU father, mother and child trio identified by respective Coriell ID NA12891, NA12892 and NA12878. This trio was deeply sequenced by Illumina with paired end reads of 100bp separated by average insert sizes of repectively $\sim$155bp, $\sim$140bp and $\sim$92bp. For NA12891, NA12892 and NA12878, there was respectively $\sim$2.5Gb, $\sim$2.7Gb and $\sim$2.9Gb of sequence data produced only for chromosome 20. After mapping the reads to reference genome HG18 we ended up with a depth of coverage of respectively $\sim$39x, $\sim$42x and $\sim$44x only for mapped reads. The 3 corresponding BAM files were sorted using samtools

Nature Methods: doi:10.1038/nmeth.2307

v0.1.17, and prepare for variant calling in 3 steps using GATK v1.4-11: (1) reads were realigned locally around indels previously discovered by the 1000 Genomes Project, (2) duplicate reads were removed, and (3) base quality scores were recalibrated separately for each lane. For variant calling, we only considered the 824,876 SNP positions reported for chromosome 20 in the interim release of the 1000 Genomes project of June 2011, and used samtools v0.1.17 to calculate Genotype Likelihoods only at those sites. Then, we control the quality of the sites using filters used in the pilot phase of the 1000 Genomes Project:

1. The depth of coverage combined across all trio members at a site must be between 50% ($\sim$ 60x) and 150%($\sim$ 200x) of the average depth, and the root mean squared mapping quality score of covering reads is at least 30 (RMS $\geq$ 30).

2. The posterior probability for at least one non-reference allele exceeds 0.999 (SNP call quality score $\geq$ 30).

3. The alleles at a site must be consistent with Mendel inheritance rules across the trio. When the 3 individuals are not all heterozygous, the phase was determined and the site was flagged as phased.

We then merged the resulting trio genotype dataset with the 1094 individuals of the 1000 Genomes project of June 2011 giving thus a dataset containing 1097 individuals defined on 731,749 SNPs. We then extracted the 384 individuals with European ancestry (populations CEU, FIN, GBR, IBS and TSI) and kept only polymorphic sites, resulting in a set of 294,658 SNPs. On this dataset, we measured phasing accuracy by first excluding the child NA12878, then proceeding with phasing using SHAPEIT or BEAGLE and finally identifying switch errors at sites previously phased using Mendel inheritance rules (cf. step3 above).

Nature Methods: doi:10.1038/nmeth.2307

ID	Population	Group	#individuals
ACB	African Caribbean in Barbados	AFR	75 (44)
ASW	African ancestry individuals from SW US	AFR	65 (44)
MKK	Maasai individuals from Kenya	AFR	31 (31)
LWK	Luhya individuals	AFR	99 (1)
YRI	Yoruba individuals	AFR	105 (99)
CLM	Colombian in Medellin, Colombia	AMR	72 (69)
MXL	Mexican individuals from LA California	AMR	70 (59)
PEL	Peruvian in Lima, Peru	AMR	70 (68)
PUR	Puerto Rican in Puerto Rico	AMR	76 (70)
CDX	Chinese Dai in Xishuangbanna, China	ASN	100 (0)
CHB	Han Chinese in Beijing	ASN	100 (100)
CHD	Chinese in metropolitan Denver, CO	ASN	1 (0)
CHS	Han Chinese South	ASN	100 (100)
GIH	Gujarati India individuals from Texas	ASN	93 (0)
JPT	Japanese individuals	ASN	100 (0)
KHV	Kinh in Ho Chi minh City, Vietnam	ASN	97 (40)
CEU	CEPH individuals	EUR	102 (4)
FIN	Finnish individuals from Finland	EUR	100 (0)
GBR	British individuals from England and Scotland	EUR	100 (1)
IBS	Iberian populations in Spain	EUR	98 (97)
TSI	Toscan individuals	EUR	100 (0)

Table S1 Detailled description of the different populations included in the Omni 2.5 dataset. The first column gives the acronym used to called the population. The second column gives the ethnical and geographical origin of the populations. The third column the continental group that were assign to each population. And finally, the fourth column gives the number of founder individuals in each population and between brackets, the number of individuals for which phase is partially known using Mendel rules.

Nature Methods: doi:10.1038/nmeth.2307

11 Supplementary Note 2 - Experiments and Results

We have carried an extensive set of analyses using a variety of large sample, whole chromosome datasets to investigate the performance of SHAPEIT2 in comparison with SHAPEIT1 and other methods. We used real datasets from SNP genotyping chips with low, medium and high SNP density, as well as SNP genotype data derived from low-coverage sequencing. Details of all the datasets are given in the previous section.

11.1 Settings used when running other methods

SHAPEIT1 SHAPEIT1 is available from
`http://www.shapeit.fr/`.
SHAPEIT1 v1.r532 was run with a total of 70 MCMC iterations: 10 burn-in (–burn 10), 10 pruning (–prune 10) and 50 main (–main 50).

Impute2 Impute2 is available from
`https://mathgen.stats.ox.ac.uk/impute/impute_v2.html`.
Impute2 v2.1.2 was run using the phase mode (-phase) with a total of 70 MCMC iterations: 20 burn-in (-burnin 20) and 50 main (–iter 70).

MaCH MaCH is available from
`http://www.sph.umich.edu/csg/abecasis/MACH/index.html`.
In all experiments, MaCH v1.0.18 was run using the phase mode (–phase) with 70 MCMC iterations (–rounds 70).

Beagle Beagle is available from
`http://faculty.washington.edu/browning/beagle/beagle.html`.
Beagle v3.3.2 was run using default parameters.

fastPHASE Fastphase is available from
`http://stephenslab.uchicago.edu/software.html#fastphase`.

Nature Methods: doi:10.1038/nmeth.2307

FastPHASE v1.4 was run using default parameters.

HAPI-UR HAPI-UR is available from
`http://code.google.com/p/hapi-ur/`.
HAPI-UR v1.01 was run as advised by the authors, that is:

1. by using a window size value (`-w`) matching the SNP density of the dataset; 64 for the European X Chromosome dataset (300K chips), 73 for the WTCCC1 derived datasets (500K chip), 79 for the Vietnamese chromosome 22 dataset (650K chip), and 110 for the WTCCC2 derived dataset (1M chip). For the 1000 Genomes Illumina Omni2.5 dataset (2.5M SNPs), the recommended value (w=184) causes the program to crash so we gradually reduced the window size value until it works and used 150.

2. by repeating the haplotype estimation three times for each dataset and making a concensus of the resulting haplotype etimates using the `vote-phase` tool provided on the HAPI-UR website.

11.2 Optimal window size

The performance of the "surrogate family" model of SHAPEIT2 is controlled by two parameters: the size of the window (W) and the number of conditioning haplotypes used in each window (K). It is well established that increasing K increases accuracy. We used the WTCCC2 X chromosome dataset to investigate how changing W influences the performance on typical GWAS data. We applied SHAPEIT2 with a fixed value of $K = 100$ and different values of W ranging from 100kb to 20Mb and measured accuracy for each W value using SER. The results shown in Fig S1 suggest that the optimal value for GWAS data is about 2Mb window. We can see a clear decrease in SER from $W = 100kb$ to W=2Mb, then a slow increase from W=2Mb to W=20Mb. In the following experiments on GWAS based datasets, we applied SHAPEIT2 using W = 2Mb.

Nature Methods: doi:10.1038/nmeth.2307

11.3 Low and medium density SNP microarray datasets

We compared methods using three datasets consisting of whole chromosomes of data from real genome-wide SNP chips typed in sample sizes representative of real GWAS studies. The first dataset consists of male X chromosome data from several European studies which were randomly paired to create 925 unrelated phase-known diploid samples with 7,821 SNPs. The second dataset consists of 740 samples with 29,227 SNPs from the Affymetrix6.0 array using randomly paired male X chromosome data from the Wellcome Trust Case Control Consortium (WTCCC2) control dataset. The third dataset consists of 1,229 phase-known samples with 7,985 SNPs on chromosome 22 of the Illumina 660K array created from a set of Vietnamese father, mother and child trios

We applied the methods SHAPEIT1, SHAPEIT2, Impute2, MaCH, Beagle and Fastphase to all three datasets. For the SHAPEIT methods, Impute2 and MaCH we varied the number of conditioning states used by each method (K), although we did not run MaCH and Impute2 for $K > 200$ on the WTCCC2 and Vietnamese datasets as these methods would have taken too long to run. It is important to note at this stage the number of conditioning states (K) used by each method has a different interpretation. In SHAPEIT2 it is the number of "surrogate family" haplotypes chosen locally in each of the windows. In Impute2 it is the number of "surrogate family" haplotypes chosen across the whole chromosome. In MaCH it is the number of random conditioning haplotypes chosen across the whole chromosome at each iteration. In SHAPEIT1 the full set of possible conditioning haplotypes are collapsed locally into a graph with K states. Beagle and Fastphase do not have such a parameter so were run using their default settings. We ran all methods on the whole chromosome at once.

The results in Figure 1 (a-b) and Supplementary Figures 1-2 show that SHAPEIT2 produces a clear increase in accuracy over SHAPEIT1 and the other approaches. On the European X chromosome (a) dataset, SHAPEIT2 with W=2Mb produces a MSD of just under 1Mb using just $K = 50$ conditioning states while SHAPEIT1 and Impute2 needs $K = 300$ conditioning states to produce the same level of accuracy. Due to the linear scaling in K of the SHAPEIT algorithms this means that on this dataset SHAPEIT2 with

Nature Methods: doi:10.1038/nmeth.2307

W=2Mb is at least 6 times faster than SHAPEIT1 for the same level of accuracy. On the WTCCC2 X chromosome dataset (b) with $K = 200$ conditioning states SHAPEIT1 produces a MSD of 1.17Mb and increases to 1.25Mb when using SHAPEIT2 with W=2Mb. This is an increase of 7%. The MSD for Impute2, MaCH, Beagle and fastPHASE are 1Mb, 0.96Mb, 0.86Mb and 0.75Mb respectively. On the Vietnamese dataset (c) with $K = 200$ conditioning states SHAPEIT1 produces a MSD of 487kb and increases to 597kb when using SHAPEIT2 with W=2Mb. This is an increase of 23%. The MSD for Impute2, MaCH, Beagle and fastPHASE are 416kb, 380kb, 360kb and 292kb respectively. When increasing the size of the window to W=5Mb (Supplementary Figures 1-2), we can see that accuracy increases in all cases on low to medium density datasets confirming the accuracy pattern obtained in Fig S1.

11.4 Performance as sample size increases

We also investigated how performance changes with increasing number of samples. We created three test datasets by combining 60 CEU HapMap2 samples with (a) 1480 control samples from the 1958 Birth Cohort, (b) 2938 samples from the 1958 Birth Cohort and UK Blood Service cohort, and (c) 8,837 samples from the 1958 Birth Control, UK Blood Service, Type 1 Diabetes, Type 2 Diabetes and Hypertension cohorts. Only SNPs on chromosome 10 and on both the Affy 500k chip used in the WTCCC1 study and in HapMap2 were combined together. The alleles at all SNPs were reported relative to the + strand of the human reference sequence. This resulted in a dataset with 23,143 SNPs. The 60 HapMap2 samples are parents of mother-father-child trios that have highly accurate haplotype estimates from trio-based phasing [11]. We ran SHAPEIT1, SHAPEIT2 with $W = 5$Mb and $W = 2$Mb and Beagle on these datasets and compared the haplotype estimates of the 60 CEU samples to the trio-based estimates of these individuals. In this way we can assess the impact of dataset size on accuracy and investigate dependence of accuracy on the combination of samples size, number of conditioning states and window size. The results in term of SER and MSD are shown in Figure 1d and Figure S2.

The results show that SHAPEIT2 with both $W = 5$Mb or $W = 2$Mb outperforms Beagle

Nature Methods: doi:10.1038/nmeth.2307

on all three datasets for all values of $K \geq 50$ and outperforms SHAPEIT1 for all values of K. There seems to be less benefit in increasing K beyond 200 for SHAPEIT2. The results of both SHAPEIT2 and Beagle improve as sample size increases but the accuracy of SHAPEIT1 seems to decrease as sample size increases, although this effect is reduced if a large value of K is used.

11.5 Using a reference set of haplotypes

We also investigated whether it is possible to increase performance by adding an external reference set of haplotypes to the set of possible conditioning haplotypes used at each stage of the SHAPEIT2 algorithm. To do this we worked with the Vietnamese dataset on chromosome 22. We took subsamples of this dataset of sizes 10, 20 50, 100, 200, 500 and 1000 and phased these datasets using three different ways of running the SHAPEIT methods. Firstly, we took a very simple approach in which we estimated the haplotypes using just the reference set of haplotypes and ignoring all the other unphased samples in the dataset. This method requires just one iteration and we use the SHAPEIT1 ($K = 200$). model to do this. This approach is very quick but does not use all of the available information in the whole dataset as it cannot use the information in other samples. Secondly, we ran SHAPEIT2 ($K = 200$, W=5Mb) on the Vietnamese samples and allowed the method to use the external haplotypes. We used the latest release of the 1000 Genomes haplotypes (Feb 2012) as the external reference set. When updating an individuals haplotype estimates we allow the haplotype subset selection method choose from the combined set of haplotypes from all other samples and the 1000 Genomes haplotypes. In addition, we initialise the phase of the Vietnamese samples using the haplotype estimates from our simple and quick method described above. This initialisation replaces the 10 burn-in iterations that we would normally have used. Thirdly, we ran SHAPEIT2 ($K = 200$, W=5Mb) without using an external reference panel. For each of the three methods we measured the SER of the estimated haplotypes.

Supplementary Figure 5 shows the results of our assessment of whether using an external reference set of haplotypes can increase accuracy. The plot shows that when the number

Nature Methods: doi:10.1038/nmeth.2307

of unphased samples is low (less then 100) then the addition of the 1000 Genomes set of haplotypes (red line) can improve accuracy relative to the approach of not using the external reference (black line). As the sample size goes above 100 we find that the approach of using the 1000 Genomes reference dataset does slightly worse than the approach of not using the reference set. The fast approach of phasing each sample one at time using only the external reference set (blue line) does better than phasing the dataset without using an external reference sample sizes lower than 50 but does not quite reach the levels of accuracy of achieved by the full MCMC approach that uses the external reference.

There are at least two factors that might explain why use of an external reference set slightly degraded performance above a sample size of 100. First, the 1000 Genomes reference set of haplotypes are derived from low-coverage sequencing data and have been constructed using imputation and may contain allele errors and phasing errors. These errors might have a downstream effect on the phasing of the Vietnamese samples if these haplotypes are included in conditioning set chosen by the "surrogate family" phasing approach.

Second, in general it makes sense that when phasing a Vietnamese sample the best set of conditioning haplotypes to use will be other Vietnamese haplotypes as they have the best chance of containing shared haplotype diversity with the sample. When presented with a set of 1000 Genomes haplotypes and a set of current haplotype estimates of other Vietnamese samples our method may choose a mixture of haplotypes from these two sources, when it might have been better to choose just from the Vietnamese set. Since we initialize the haplotype estimates of the Vietnamese samples using only the 1000 Genomes reference set it may be that this causes a bias that is difficult to correct in subsequent iterations. Of course, the genome is large and in a given region the converse might be true. It might be the case that the ancestry of a given Vietnamese sample is closer to some of the 1000 Genomes samples so that haplotype estimation is improved by using the external reference. It could be quite difficult to precisely determine the reasons for the small differences we have observed and would require more space and analysis than we currently have in this paper. In scenarios where the ancestry of the samples being phased is less clear cut the use of an external reference could be a good approach.

Nature Methods: doi:10.1038/nmeth.2307

The approach of phasing each sample one at a time using just the external reference did have good performance in small samples and since it is very quick (approximately 70 times faster than our full MCMC approach) it might be of use to researchers working with small sample sizes to obtain a quick "first pass" set of haplotypes.

11.6 High density SNP microarray dataset

We carried out a methods comparison using the data from a denser Illumina Omni2.5 SNP microarray. The 1000 Genomes Project have used this chip to obtain genotypes on 2,123 samples across 14 different populations. These samples allow us to assess the performance of our method when applied to datasets that contain samples with different ancestry. The dataset consists of a mixture of unrelated samples, mother-father-child trios and parent-child duos. We took the 1,754 founder individuals across the whole dataset and applied SHAPEIT1, SHAPEIT2 with both $W = 5$Mb and $W = 2$Mb, Beagle and HAPI-UR(3x). We only used these three methods as they are the only methods that can handle such high-density datasets across whole chromosomes in reasonable amounts of computing time. The methods were run only on data from chromosome 20 which consisted of 54,267 SNPs. To assess the method we took the haplotype estimates from the 696 samples that were parents of either trios of duos in the full dataset and compared them to haplotype estimates dictated purely by Mendel rules. A large proportion of sites will have phase determined in this way and provide an unbiased set of true haplotypes for methods comparison. Sites that did not have their phase determined by Mendel rules were excluded when calculating the switch errors of the estimated haplotypes. The results are shown in Figure 1c and Supplementary Figure 2. We find that SHAPEIT2 produces a large boost in accuracy over SHAPEIT1 and Beagle. With $K = 200$ conditioning states and W=5Mb SHAPEIT2 (MSD = 400kb, SER = 1.9%) increases accuracy by 30% relative to SHAPEIT1 (MSD = 308kb, SER=2.4%), increases by 39% relative to Beagle (MSD = 287kb, SER=2.6%) and increases by 40% relative to HAPI-UR(3x) (MSD = 284kb, SER=2.6%) When using smaller window sizes of $W = 2$Mb instead of $W = 5$Mb, SHAPEIT2 gets slighlty better results (MSD = 421kb, SER = 1.8%).

Nature Methods: doi:10.1038/nmeth.2307

11.7 Haplotype estimation in sequencing studies

We wanted to evaluate the performance of our new method when SNP density reaches near maximum levels, as would be the case if SNP genotypes were derived by sequencing. To do this we considered SNP genotype calls made from the Phase I of the 1000 Genomes Project on chromosome 20 and designed the two following distinct experiments.

11.7.1 Comparison to phase known sites on the Illumina Omni2.5M dataset

The 1,094 Phase I individuals are all included in the Illumina Omni2.5M dataset (see previous subsection). Usefully 418 of them have also relatives in the larger Omni2.5 dataset of 2,123 samples, allowing us to determine true phase for them using Mendel rules at sites that are in both Phase I and Omni2.5. We re-phased the 1,094 individuals over 12 Mb of the chromosome 20 (39Mb to 51Mb) using SHAPEIT2 with a grid of parameter values where the size of the window ranges from 100kb to 5Mb and the number of conditioning haplotypes ranges from 100 to 500. We applied Beagle as well on this dataset using default settings. Additionally, we extracted, for the same 12Mb region, the haplotypes released by the 1000 Genomes Project. To avoid any edge effect, we measured phasing accuracy only in the central 10Mb region. The results are shown in Figure S3. We find that the best window size for sequencing data seems to be comprised between W=0.2Mb and W=0.4Mb, and SHAPEIT2 with $K = 100$ reaches the accuracy of the haplotypes released by the project using windows of this size. SHAPEIT2 with $K = 200$ and $K = 500$ provide even better haplotypes as soon as the windows are smaller than respectively 2Mb and 5Mb. Beagle produces substantially lower quality haplotypes than the official 1000 Genomes haplotypes and the best SHAPEIT2 runs. All these results suggest that the best window size value is W=0.3Mb.

These analyses suggest that a more accurate set of 1000 Genomes haplotypes could be obtained by re-phasing the 1000 Genomes Phase 1 genotypes using SHAPEIT2 with a high value of K.

Nature Methods: doi:10.1038/nmeth.2307

11.7.2 Comparison to phase known sites in a trio sequenced at high-coverage

In a second experiment, we took the SNP genotypes from the 381 European samples in the Phase 1 dataset and combined it for the whole chromosome 20 with the genotypes of the parents of a mother-father-child trio sequenced at high-coverage (∼130x across all three individuals) by Illumina. We only used genotypes at the non singletons SNPs in the Phase 1 dataset. This resulted in a dataset with 294,658 SNPs that were polymorphic in the 383 samples. We estimated haplotypes from this dataset using SHAPEIT2 with $K = 100$ conditioning haplotypes and windows of W=0.3Mb, W=0.5Mb, W=1Mb, W=2Mb and Beagle using default settings. Then, we compared the estimated haplotypes of the trio parents to a set of haplotypes derived using family information from the high-coverage genotype calls. The results are shown in Supplementary Table 1. We find that SHAPEIT2 with W=0.3Mb increased the MSD compared to Beagle by $\sim 62\%$ (SHAPEIT2 MSD=187kb, Beagle MSD=115kb).

11.8 Performance on multi-population datasets

Genowide SNP genotypes are often collected from multiple populations, especially for the purposes of population genetic analyses. The Illumina Omni2.5M dataset is a perfect example of such study design since it contains individuals from 14 populations (Table S1). We investigated the performance of each method stratified by population. All individuals were phased together using SHAPEIT2 with $K = 200$ and W=5Mb, SHAPEIT1 with $K = 200$ and Beagle. Then, we measured SER on the subset of 696 individuals for which phase can be determined using Mendel rules, and regrouped the individuals according to the 10 distinct populations they belong to. Note that some populations were discarded if the number of phase known individuals was less than 10. The results are shown in Supplementary Figure 7. SHAPEIT1 performs equally well as Beagle for most of the populations but substantially outperforms Beagle in populations with some proportion of African ancestry (YRI, ASW, ACB). SHAPEIT2 outperforms clearly both SHAPEIT1 and SHAPEIT2 for all populations. For African populations, Beagle, SHAPEIT1, and SHAPEIT2 give a SER of 2.7%, 1.9% and 1.5% respectively.

Nature Methods: doi:10.1038/nmeth.2307

A question that often arises in this context of multi-population datasets concerns the phasing strategy to be adopted in order to get the best possible set of haplotypes; should all populations be phased simultaneously (scenario A) or separately according to some predefined continental groups (scenario B)? The advantage of using the largest sample size possible is that this is known to increase phasing accuracy. However, it is also known that LD patterns can vary substantially across different populations and thus could be a confounding factor for phasing. We assessed this question on the Illumina Omni2.5M dataset using SHAPEIT2. To mimic scenario A, we phased the 1754 individuals all together using $K = 100$ conditioning haplotypes, windows of W=5Mb and without any ancestry consideration. Then, we measured phase accuracy on the 697 duo/trio parents for which very accurate phase can be determined through Mendel rules. To mimic scenario B, we first classified the 1754 individuals in the following 4 majors ethnic groups : African (AFR), American (AMR), Asian (ASN) and European (EUR) containing respectively 375, 288, 591 and 502 individuals (Table S1). Then, we phased each group separately using the same model parameters ($K = 100, W$=5Mb) and we measured phasing accuracy for respectively 189, 267, 141 and 104 trio/duo phased individuals. The results are shown in Supplementary Figure 6 where we plotted the per-individual switch error percentages obtained for scenario B against those obtained for scenario A. Individuals are coloured according to the group (AFR, AMR, ASN and EUR) they belong to. Overall, phasing accuracy is substantially reduced when phasing populations all together. When examining the phasing accuracy per individual group, one can notice that the improvement is mainly achieved for the AMR group, in a lesser extent for the EUR and AFR groups and not at all for the ASN group. This suggests that increasing the size of the sample should be prioritized when using SHAPEIT2 even though individuals do not always share common ancestries. Moreover, this illustrates three main properties of the surrogate family model used in SHAPEIT2. First, it gets rid of any confounding effect due to mixing different LD patterns, as illustrated by the fact that none of the groups shows an increased error rate. That means that in the worst case of scenario A, the same conditioning haplotypes are chosen as in scenario B. Second, it can slightly improve accuracy in some cases as illustrated by the results obtained for AFR and EUR. This is

Nature Methods: doi:10.1038/nmeth.2307

probably due to the increasing size of the set of haplotypes from which the conditioning haplotypes are chosen. And third, it can even greatly improve accuracy when dealing with admixed individuals as illustrated by results for the AMR group. This group contains individuals with mixed ancestries of Native American, African and European. In this particular case, the conditioning haplotypes in a window are chosen according to the local ancestry of this window, thus making use of haplotypes of other AMR individuals but also of European (EUR) and African (AFR) origins when required.

11.9 Comparison with partition-ligation strategies

An alternative strategy for phasing whole chromosomes would be to run a phasing method on overlapping regions of the chromosome. Haplotypes estimated in each region could then be joined together to form a set of haplotypes across the whole chromosome. This is the strategy recommended when using Impute2, as this method is designed to work on regions of up to around 5Mb in size. The advantage of this strategy is that it can be trivially parallelized on a compute cluster with many cores. The disadvantage is that it may be more more complicated to set up, the ligation step needs to be carried out which may cause errors. The strategy is also somewhat wasteful in the sense that the overlapping regions of the regions (which do need to be a reasonable size) are phased twice. We refer to this strategy as partition-ligation (PL). Any phasing method can be used to phase each region.

We recommend that SHAPEIT2 is used to phase whole chromosomes using one single compute job. When multiple cores are available the multi-threading option can be used to take advantage of them. This approach is much simpler to run, requiring just one compute job per chromosome, and it does not require a ligation step.

We compared the accuracy and computational burden of three strategies (a) PL using Impute2 (K=100) (b) PL using SHAPEIT2 (K=100) (c) SHAPEIT2 run as one job per chromosome. We applied these methods to the European chromosome X dataset, the WTCCC2 chromosome X dataset, the Vietnamese chromosome 22 dataset and the 1000 Genomes Illumina Omni2.5 dataset. For both the PL strategies, we split the datasets

Nature Methods: doi:10.1038/nmeth.2307

into 5Mb chunks across the chromosome and created overlaps between the chunks by extending both boundaries by 0.25Mb. Then, we phased each resulting 5.5Mb regions and ligated consecutive chunks in the following way.

Let h_{ilk} denote the jth haplotype of individual i fr For each pair of consecutive chunks in each individual we considered the overlapping haplotypes

Let $h_l = \{h_{l1}, h_{l2}\}$ and $h_{l+1} = \{h_{(l+1)1}, h_{(l+1)2}\}$ denote the two haplotypes from the overlapping region between chunks l and $l+1$. We measure the hamming distance between haplotypes for the two possible alignments of the two chunks and use that measure to decide which is the best alignment. More specifically, we calculate $d_1 = d(h_{l1}, h_{(l+1)1}) + d(h_{l2}, h_{(l+1)2})$ and $d_2 = d(h_{l1}, h_{(l+1)2}) + d(h_{l2}, h_{(l+1)1})$, where $d(h, g)$ is the Hamming distance between two haplotypes h and g. If $d_1 < d_2$ we choose to align h_{l1} with $h_{(l+1)1}$ and h_{l2} with $h_{(l+1)2}$, otherwise we align h_{l1} with $h_{(l+1)2}$ and h_{l2} with $h_{(l+1)1}$. The details of how the haplotypes are aligned are also important. If we decide to align h_{l1} with $h_{(l+1)1}$ then we do the alignment at the midpoint of the overlapping region i.e we exclude the 125kb at the end of chunk l and at the start of chunk $l+1$ and then join the first haplotypes from chunk l with the first haplotype from chunk $l+1$. This attempts to avoid edge effects.

The results of this analysis are shown in Figure 1f in the main paper and show that on all 4 datasets the most accurate strategy is to run SHAPEIT2 on the whole chromosome at once. The next best strategy is to run PL using SHAPEIT2. Running PL using Impute2 is the worst strategy. Interestingly, the difference in SER between the two PL strategies is almost the same as the difference between the two SHAPEIT2 strategies on all of the datasets except the Vietnamese dataset.

The difference in CPU time is as expected. Since Impute2 scales quadratically with K we see that PL with Impute2 takes longer to run than either of two SHAPEIT2 approaches. There is very little difference between the CPU time of running PL with SHAPEIT2 or running SHAPEIT2 on whole chromosomes.

Nature Methods: doi:10.1038/nmeth.2307

11.10 Comparison of Algorithm 1 and 2 for haplotype selection.

We also used the Illumina Omni2.5M dataset to compare both Algorithm 1 and 2 for selecting conditioning haplotypes. Figure S4 shows the switch error rate of each of the 696 samples using both methods. The plot shows that some individuals have very high switch error rates using the basic method implemented in Algorithm 1 and the error rate on these individuals is reduced to normal values when using Algorithm 2. Figure S5 shows the per-sample switch error rate of SHAPEIT1 versus SHAPEIT2. This figure further illustrates that SHAPEIT2 provides improves phasing accuracy compared to SHAPEIT1. SHAPEIT1 does not use a haplotype subset selection method so is relatively more immune to the problem of samples sharing long segments with an IBD count of 2. This can be seen in Figure S5 as SHAPEIT1 does not produce large switch error rates on some samples but is highly correlated with the SHAPEIT2 algorithm that uses Algorithm 2 for haplotype selection.

11.11 Impact on downstream imputation quality

A major use of phasing is haplotype estimation of GWAS samples in order to speed up imputation from large reference panel of haplotypes such as 1000 Genomes. The current recommendation is that GWAS samples are first 'pre-phased' using the most accurate method available. The subsequent imputation step (which involves imputing alleles from one set of haplotypes into another set) is fast. As new haplotype reference sets become available imputation can be re-run much more efficiently.

Given the widespread use of imputation many groups it is of interest to investigate the impact of phasing accuracy on downstream imputation. To do so, we used 2490 WTCCC2 samples that were typed with both Affymetrix 6.0 (set A of $\sim$ 1M SNPs) and Illumina 1M (set B of $\sim$ 1M SNPs) and we proceeded as follows:
(1) we extracted the genotypes at SNPs included on the Affymetrix 500k (set C of $\sim$ 500K SNPs),
(2) we phased set C on the whole chromosome 10 (23,231 SNPs) using Beagle, HAPI-UR

Nature Methods: doi:10.1038/nmeth.2307

and SHAPEIT2 ($K = 100$, $W = 2Mb$). We also phased the dataset by making 27 chunks of 5Mb and then by applying MaCH and SHAPEIT2 ($K = 100$, $W = 2Mb$) on each chunk,

(3) we imputed in each resulting set of haplotypes all chromosome 10 SNPs included in sets A and B but not in C (47,739 SNPs) using Impute2 and 1000 Genomes phase 1 release 3 haplotypes as reference panel,

(4) we measured in each scenario the imputation quality as the R^2 correlation coefficient between the imputed and the true genotypes.

The results of this analysis are shown in Supplementary Figures 3 and 4. These plots show the imputation quality differences achieved when using SHAPEIT2 instead of Beagle (Supplementary Figure 3a), HAPI-UR (Supplementary Figure 3b), MaCH (Supplementary Figure 4a) and SHAPEIT2 when run in chunks (Supplementary Figure 4b). Overall, we can see a clear trend suggesting that using SHAPEIT2 for prephasing leads to improved imputation quality especially at the hardest SNPs to impute ($r^2 < 0.8$). Moreover, running SHAPEIT2 on the whole chromosome seems to work slightly better than when running on a set of 5Mb chunks.

11.12 Computational performance

We have also compared the computational performance of the methods used in our comparison. We compared all methods using just 1 thread. In addition, to illustrate the gains that can be achieved by multi-threading we also SHAPEIT2 with 1 thread. We measured the elapsed time taken for each command to run. Figure S6 shows the time taken by each method relative to the time taken by Beagle on the European chromosome X dataset, the WTCCC2 chromosome X dataset, the Vietnamese chromosome 22 dataset and the Omni2.5 chromosome 25 dataset. Figure S7 shows the CPU time taken by each method.

On the Vietnamese chromosome 22 dataset consisting of 1,229 samples SHAPEIT2 ($K = 100, W = 5$Mb with 1 and 4 threads took 179mins and 48mins to run respectively. Beagle took 23mins and HAPI-UR (3x) took 22mins. Assuming 20 cores are available, extrap-

Nature Methods: doi:10.1038/nmeth.2307

olating these numbers to the whole genome (using the assumption that chromosome 22 is roughly 1/60th of the genome) it would take SHAPEIT2 (run with 1 thread) approximately 9 hours to phase the dataset. Both Beagle and HAPI-UR (3x) would take around 66mins.

On the Illumina Omni2.5M chromosome 20 dataset consisting of 1,740 samples SHAPEIT2 ($K = 100, W = 5$Mb) with 1 and 4 threads took 26.7 hours and 7 hours to run respectively. Beagle took 9.2 hours and HAPI-UR (3x) took 5.2 hours. Assuming 20 cores are available, extrapolating these numbers to the whole genome (using the assumption that chromosome 20 is roughly 1/45th of the genome) it would take SHAPEIT2 (with 1 thread) approximately 60 hours to phase the dataset. Beagle and HAPI-UR (3x) would take around 20.7 hours and 11.7 hours respectively.

It is our experience that many research groups working on GWAS datasets have access to compute resources equivalent to 20 cores. Under this assumption it is clear that timing would not present a serious hurdle for phasing. GWAS are very expensive experiments that will often have taken many months on data collection. The resulting haplotypes from such studies will often we used many times in subsequent analysis. For this reason it seems preferable to ensure that the haplotypes are as accurate as possible via phasing using SHAPEIT2.

When more cores are available then the multi-threading features in SHAPEIT2 become very useful. HAPI-UR (3x), which runs HAPI-UR 3 times and then combines the results, can also take advantage of 3 cores at a time when phasing a chromosome.

Figures S8 and S9 show the computational performance of SHAPEIT1, SHAPEIT2 and Beagle on the three datasets that combine HapMap CEU samples with increasingly larger amounts of WTCCC1 data. The figures show timings relative to the time taken by Beagle and CPU time respectively plotted against the number of conditioning states used by SHAPEIT1 and SHAPEIT2. These results highlights how Beagle scales non-linearly as the number of samples increases. This can be more clearly seen in Figure S9 that shows the computational time taken by the methods plotted against the sample size.

On the dataset with very high SNP density produced by sequencing from the 1000

Nature Methods: doi:10.1038/nmeth.2307

Genomes Project we found that Beagle took 6 hours whereas SHAPEIT2 ($K = 100, W = 0.3Mb$) and 4 threads took 5.8 hours (Supplementary Table 1).

Nature Methods: doi:10.1038/nmeth.2307

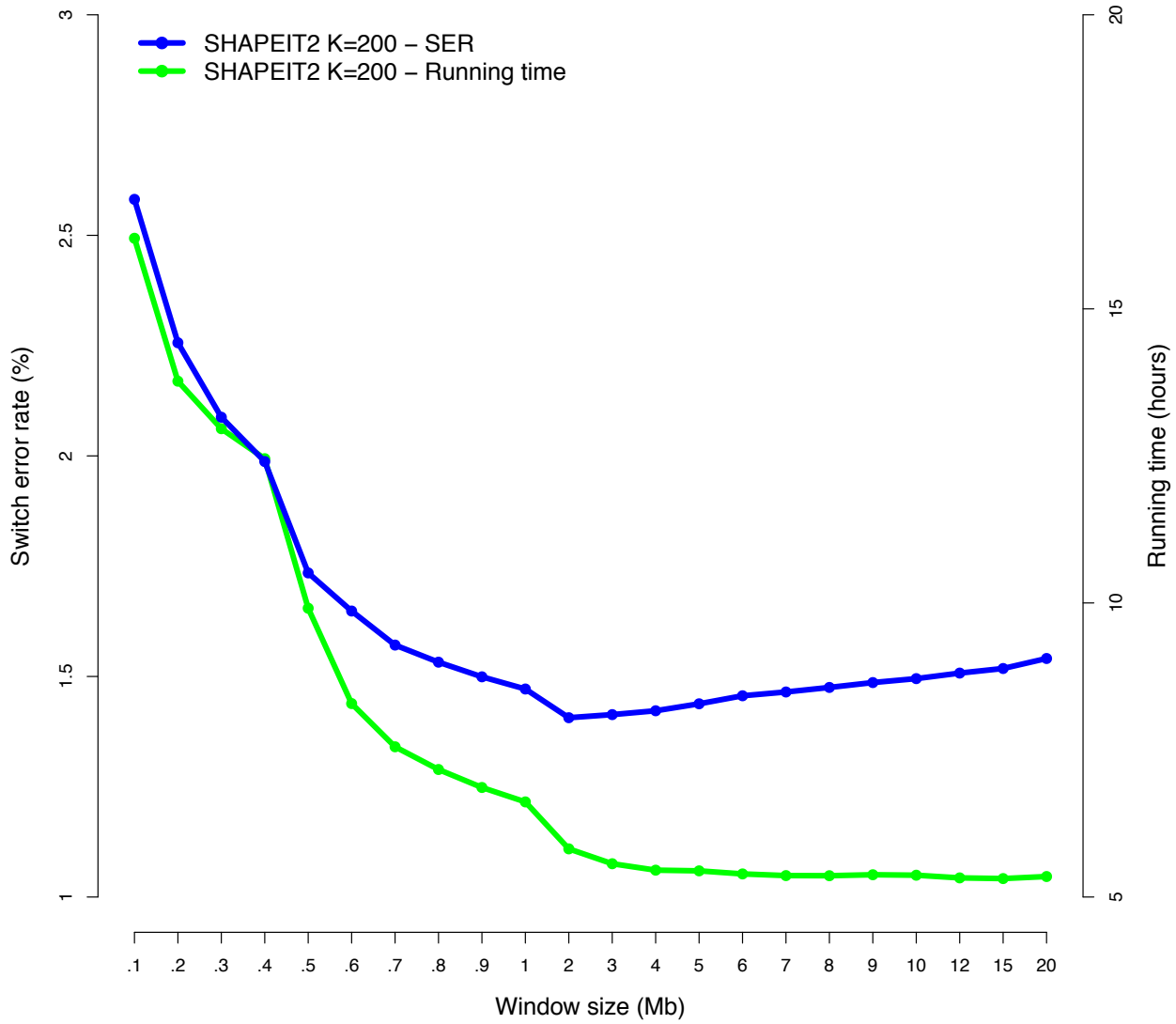


Fig S1 Impact of the window size (in Mb) on the switch error percentage (blue line) of SHAPEIT2 (K=200) and the running times in hours (green line) for the WTCCC2 X chromosome dataset. For this experiment SHAPEIT2 was run using 10 burn in iterations, 10 sampling iterations and 50 main iterations.

Nature Methods: doi:10.1038/nmeth.2307

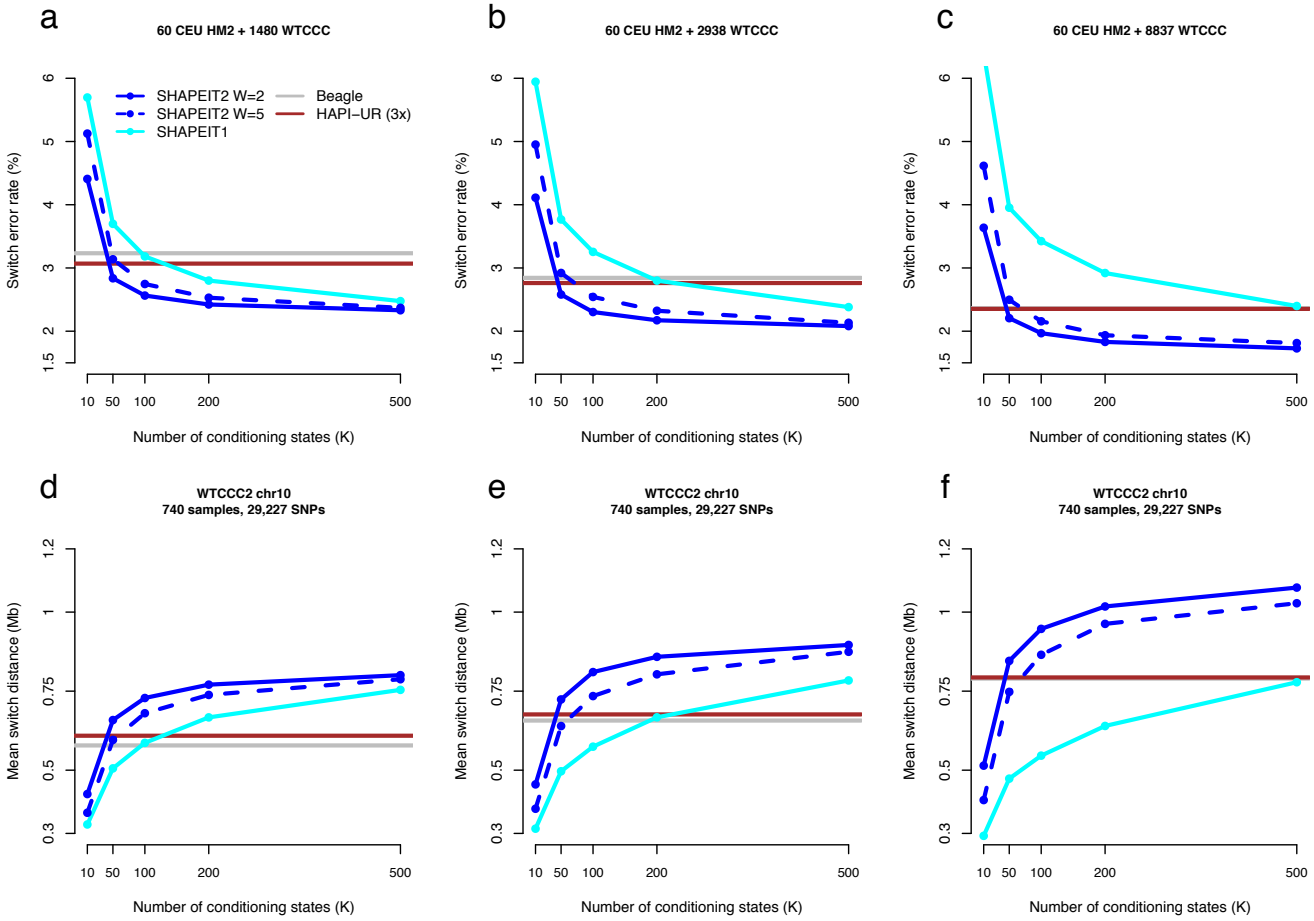


Fig S2 Comparison of the accuracy of methods on the 60 CEU samples plus 1480 (a, d), 2938 (b, e) and 8,837 (c, f) WTCCC1 samples. The datasets all have 23,143 SNPs from chromosome 10. The plots show the switch error percentage (a-c) and the mean switch distance in Mb (d-f) plotted against the number of conditioning states for SHAPEIT1 (cyan), SHAPEIT2 with W=5Mb (solid blue) and SHAPEIT2 with W=2Mb (dashed blue). Beagle (grey) and HAPI-UR (brown) were run using default settings.

Nature Methods: doi:10.1038/nmeth.2307

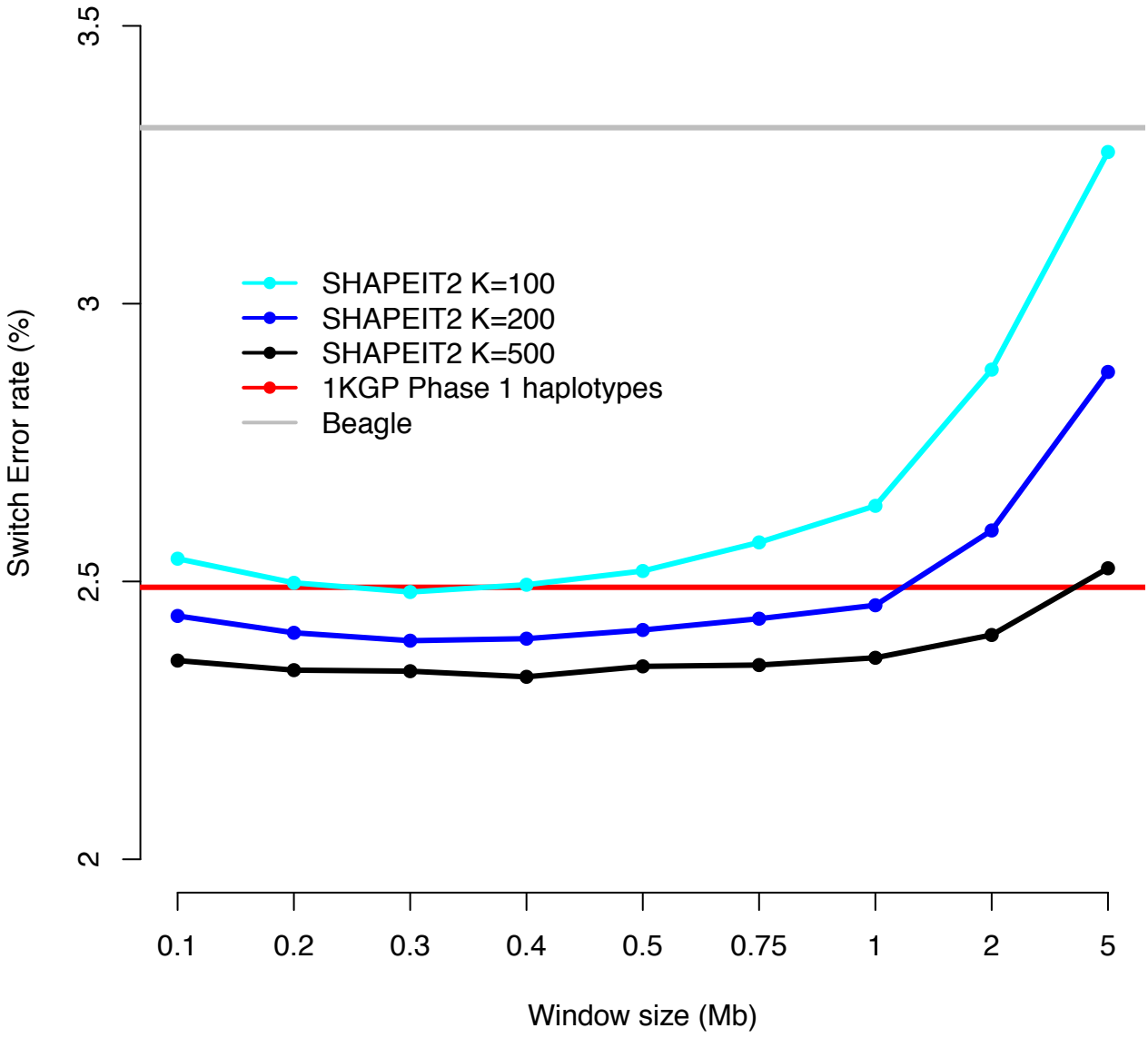


Fig S3 Comparison of methods on 10Mb of chromosome 20 of the 1000 Genomes Phase I data. Switch error percentage is plotted against the size of the window used for SHAPEIT2 with $K = 100$ (cyan), SHAPEIT2 with $K = 200$ (blue) and SHAPEIT2 with $K = 500$ (black). Beagle (grey) was run using default settings. The haplotypes released by the project were obtained using SNPTools (red).

Nature Methods: doi:10.1038/nmeth.2307

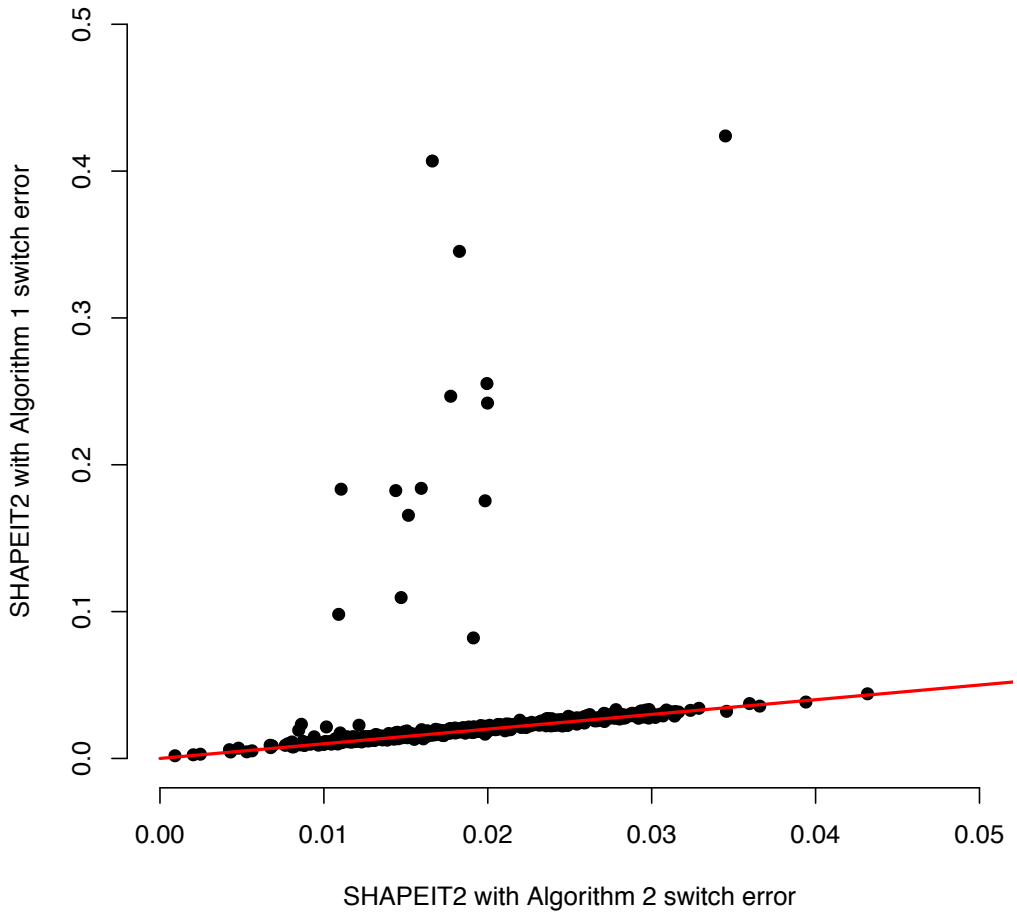


Fig S4 Comparison of a haplotype subset selection methods. Algorithm 2 (x-axis) which is robust to cryptic relatedness is compared to Algorithm 1 (y-axis) which is not robust to cryptic relatedness. The methods were applied to the Illumina Omni2.5 dataset of 1,754 samples. Per-sample switch error rates of the estimated haplotypes from the two algorithms are plotted against each other. Only the 696 samples that have family-based haplotype estimates available were used for the comparison.

Nature Methods: doi:10.1038/nmeth.2307

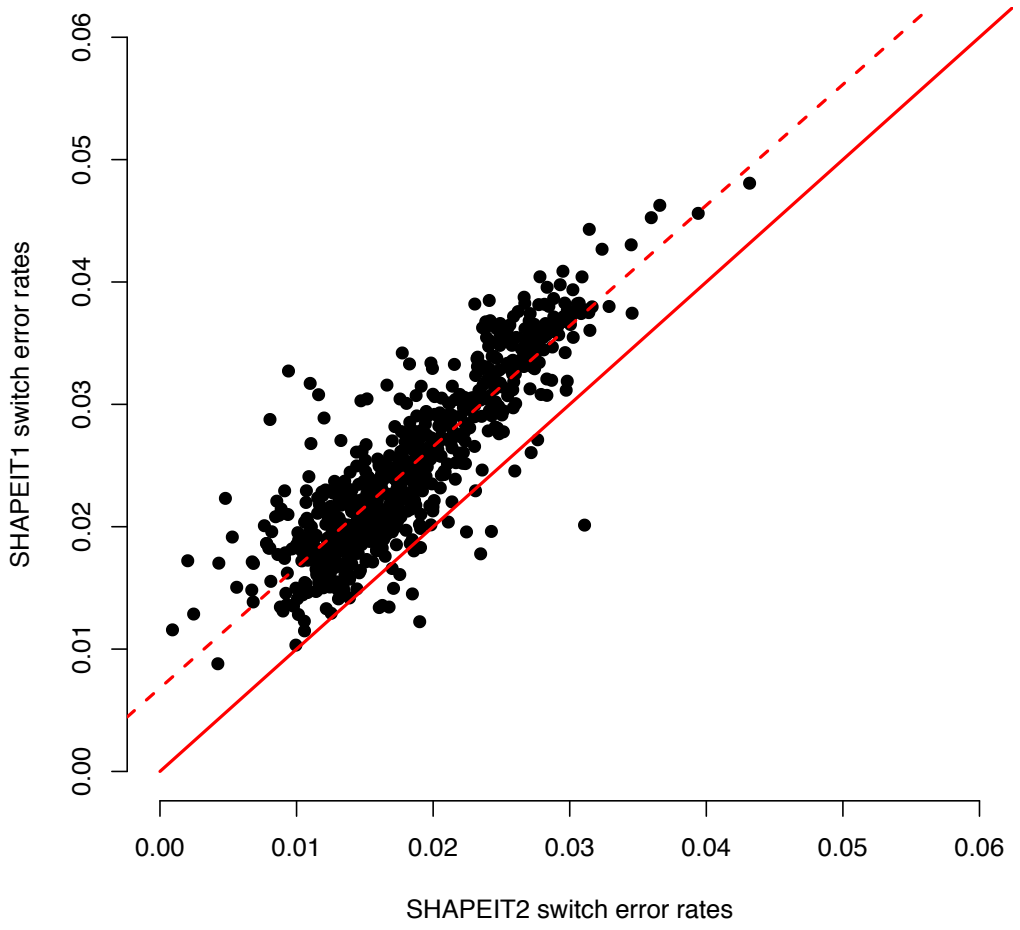


Fig S5 Comparison of per-sample switch error rate for SHAPEIT2 (x-axis) versus SHAPEIT1 (y-axis). The methods were applied to the Illumina Omni2.5 dataset of 1,754 samples. Per-sample switch error rates of the estimated haplotypes from the two algorithms are plotted against each other. Only the 696 samples that have family-based haplotype estimates available were used for the comparison. The solid red line is the x=y line. The dotted red line is the regression line.

Nature Methods: doi:10.1038/nmeth.2307

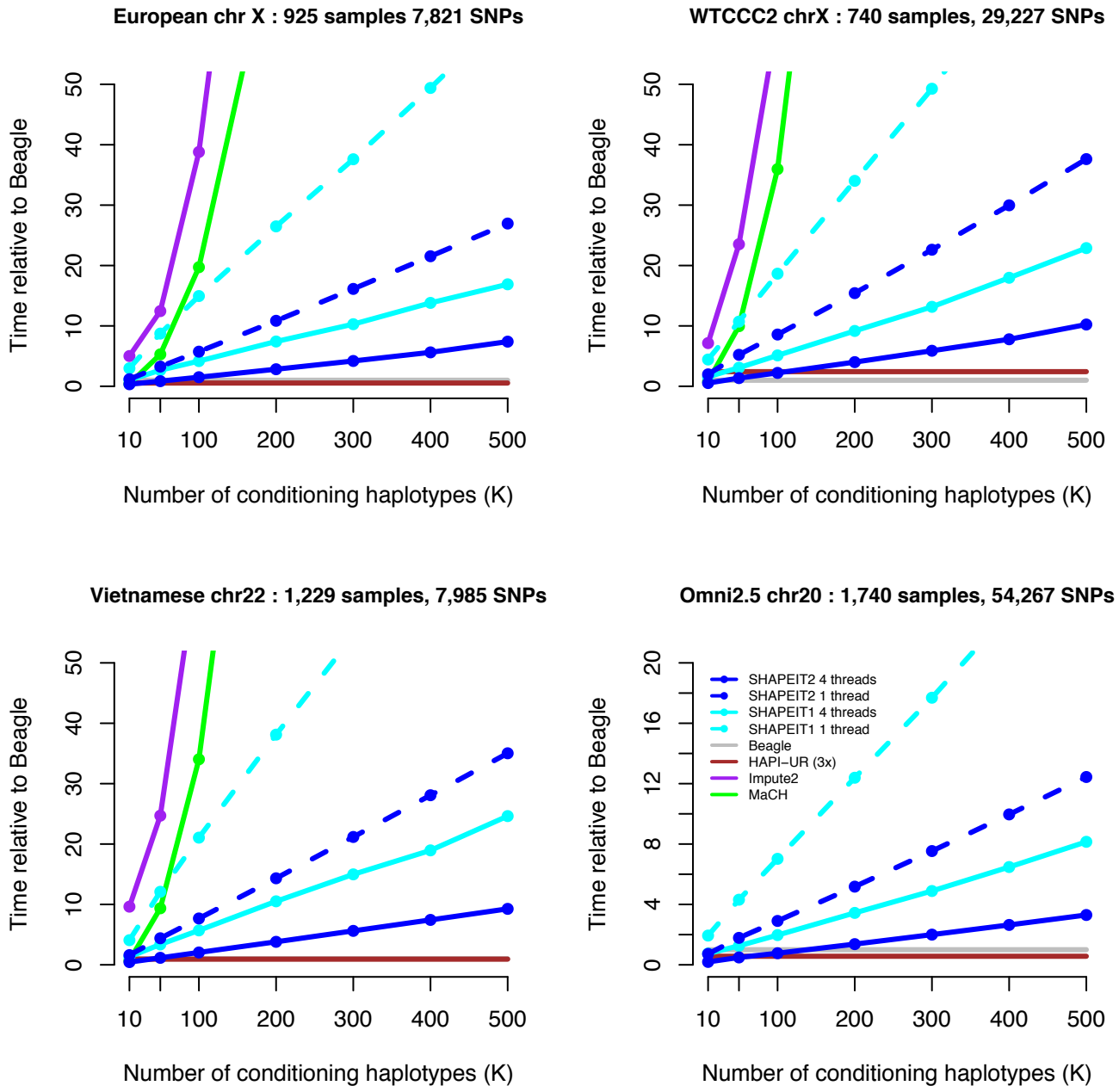


Fig S6 Comparison of the computational performance of methods on the datasets : European chromosome X (top left), WTCCC2 chromosome X (top right), Vietnamese chromosome 22 (bottom left), Omni2.5 chromosome 20 (bottom right). Computational timings relative to Beagle are plotted against the number of conditioning states (K) used by the methods. SHAPEIT2 was run with $W = 2$Mb.

Nature Methods: doi:10.1038/nmeth.2307

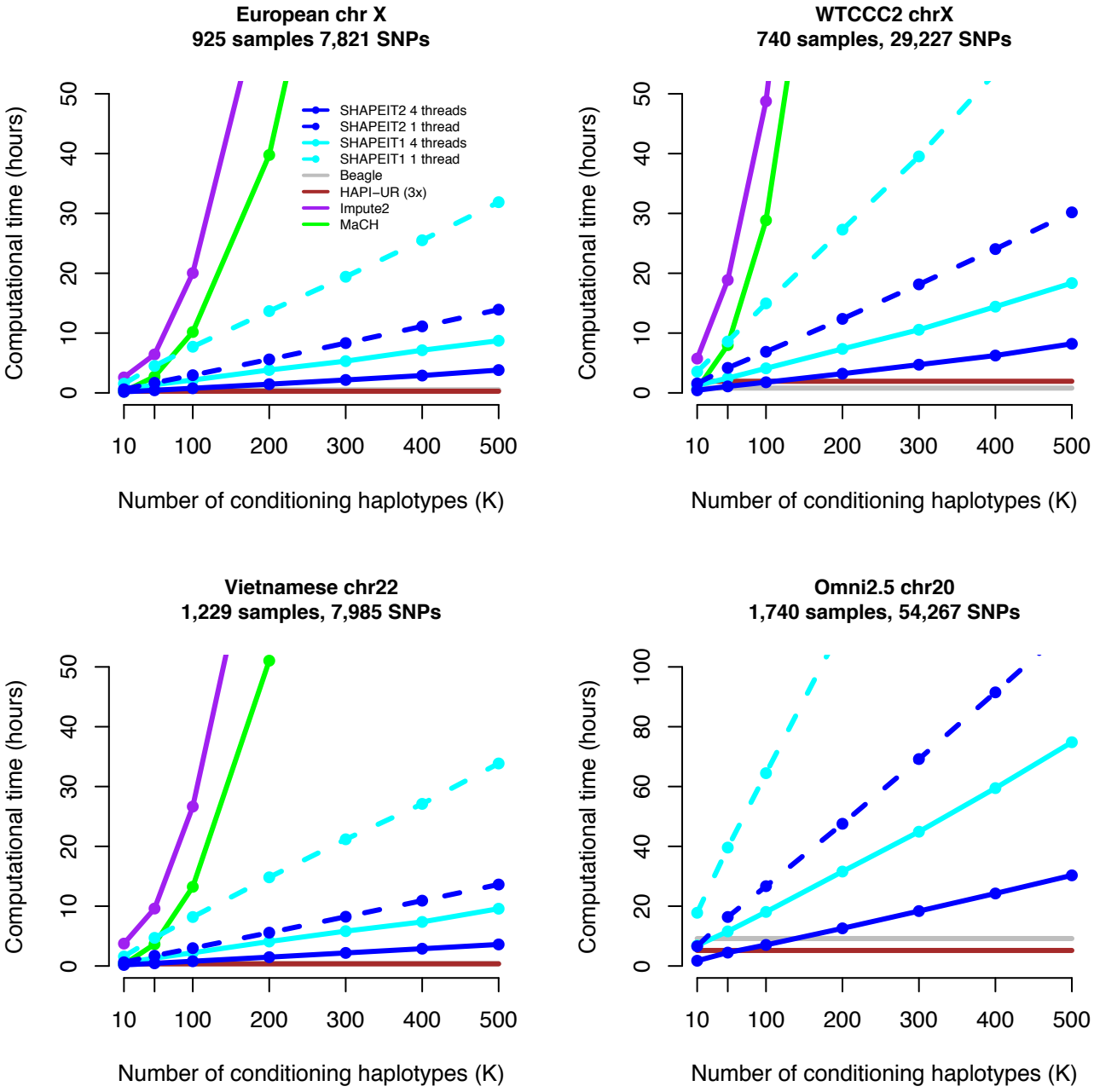


Fig S7 Comparison of the computational performance of methods on the datasets : European chromosome X (top left), WTCCC2 chromosome X (top right), Vietnamese chromosome 22 (bottom left), Omni2.5 chromosome 20 (bottom right). Computational timings in hours are plotted against the number of conditioning states (K) used by the methods. SHAPEIT2 was run with $W = 2$Mb.

Nature Methods: doi:10.1038/nmeth.2307

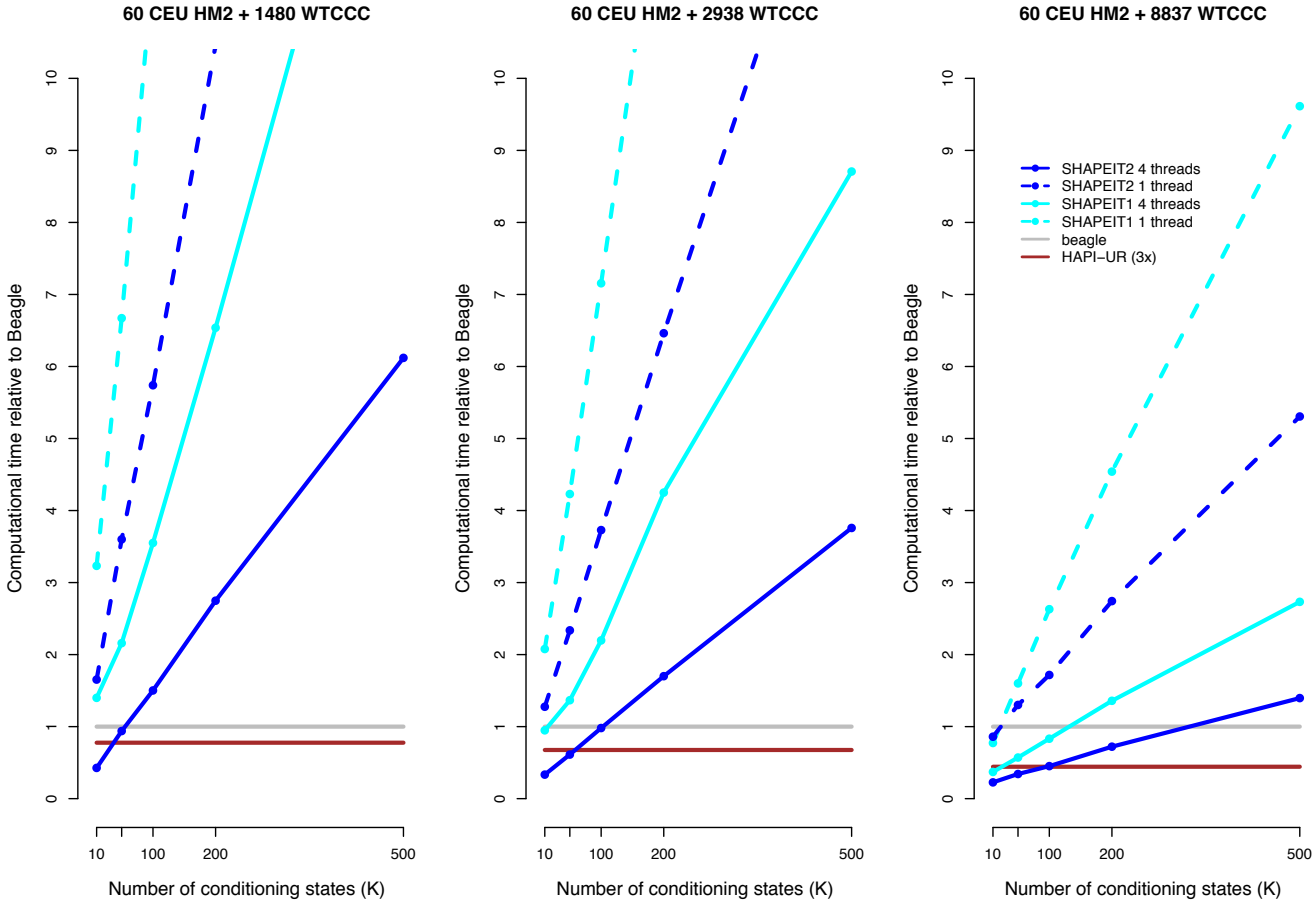


Fig S8 Comparison of computational performance of SHAPEIT1 (cyan), SHAPEIT2 (blue), Beagle (grey) and HAPI-UR(3x) (brown) on 60 HapMap2 CEU samples plus 1480 (left), 2938 (middle) and 8837 (right) WTCCC1 samples. Computational time is shown relative to Beagle. SHAPEIT2 was run with $W = 2$Mb.

Nature Methods: doi:10.1038/nmeth.2307

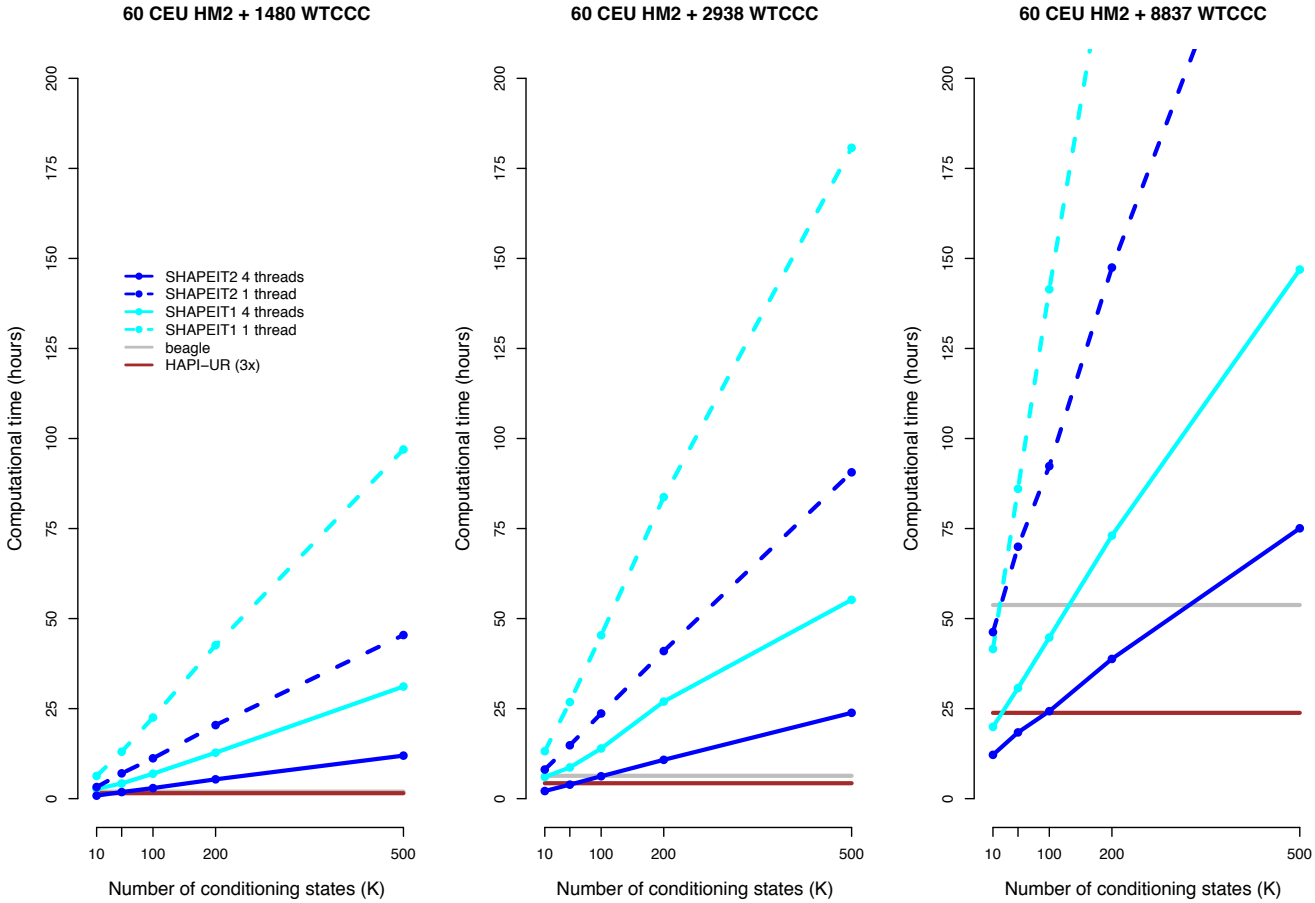


Fig S9 Comparison of computational performance of SHAPEIT1 (cyan), SHAPEIT2 (blue), Beagle (grey) and HAPI-UR(3x) (brown) on 60 HapMap2 CEU samples plus 1480 (left), 2938 (middle) and 8837 (right) WTCCC1 samples. Computational time is shown in hours. SHAPEIT2 was run with $K = 100$ and $W = 2$Mb.

Nature Methods: doi:10.1038/nmeth.2307

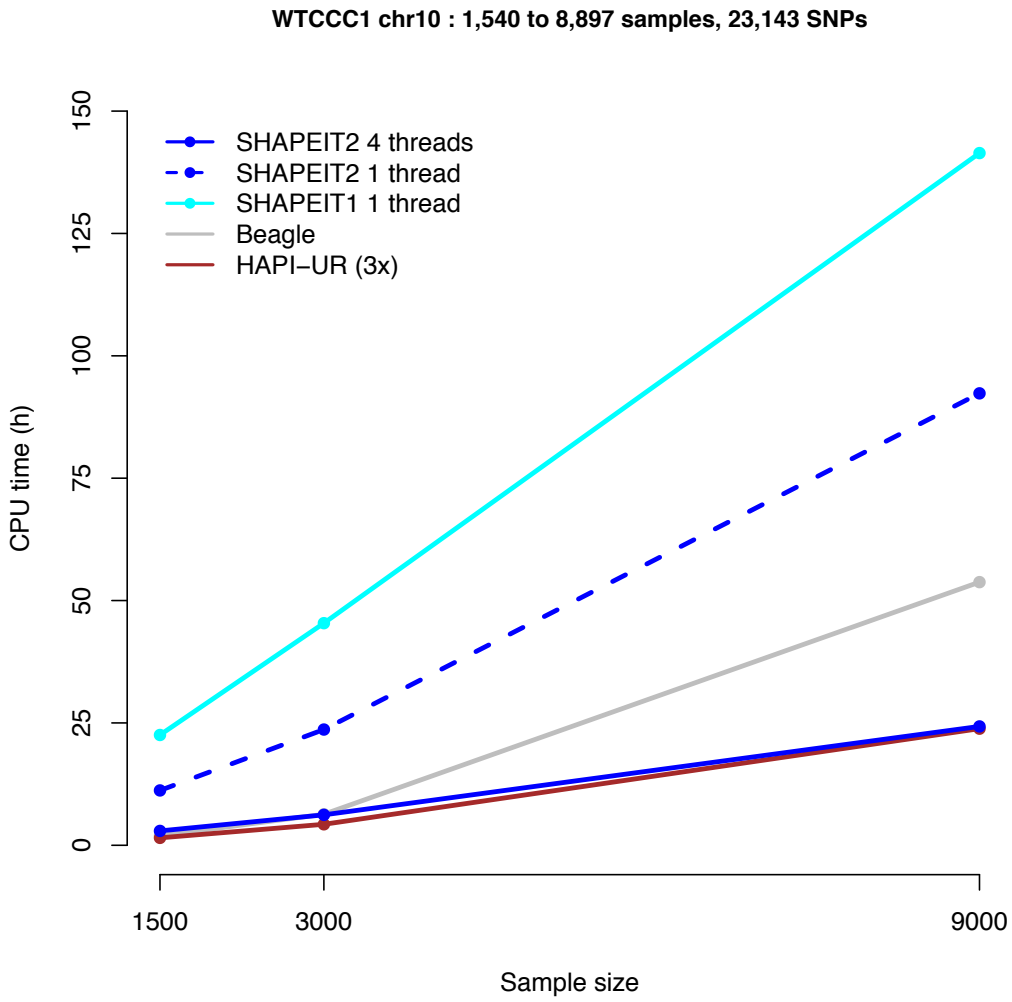


Fig S10 Comparison of computational performance of SHAPEIT1 (cyan), SHAPEIT2 (blue), Beagle (grey) and HAPI-UR(3x) (brown) on 60 HapMap2 CEU samples with varying number of WTCCC1 samples. The x-axis reports the total sample size. Computational time is shown relative to Beagle. SHAPEIT2 was run with $K = 100$ and $W = 2$Mb.

Nature Methods: doi:10.1038/nmeth.2307

References

[1] A Alcaïs, A Alter, G Antoni, M Orlova, and N Van Thuc. Stepwise replication identifies a low-producing lymphotoxin-allele as a major risk factor for early-onset leprosy. *Nature*, 39(4):517–522, 2007.

[2] A Alter, N T Huong, M Singh, M Orlova, N Van Thuc, K Katoch, X Gao, V H Thai, N N Ba, M Carrington, L Abel, N Mehra, A Alcaïs, and E Schurr. Human Leukocyte Antigen Class I Region Single-Nucleotide Polymorphisms are Associated with Leprosy Susceptibility in Vietnam and India. *Journal of Infectious Diseases*, 203(9):1274–1281, March 2011.

[3] B Browning and S Browning. A unified approach to genotype imputation and haplotype-phase inference for large data sets of trios and unrelated individuals. *American journal of human genetics*, 84(2):210–223, February 2009.

[4] Olivier Delaneau, Jonathan Marchini, and Jean-François Zagury. A linear complexity phasing method for thousands of genomes. *Nature methods*, 9(2):179–181, February 2012.

[5] Bryan Howie, Christian Fuchsberger, Matthew Stephens, Jonathan Marchini, and Gonçalo R Abecasis. Fast and accurate genotype imputation in genome-wide association studies through pre-phasing. *Nature genetics*, July 2012.

[6] Bryan N Howie, Peter Donnelly, and Jonathan Marchini. A flexible and accurate genotype imputation method for the next generation of genome-wide association studies. *PLoS genetics*, 5(6):e1000529, June 2009.

[7] Augustine Kong, Gisli Masson, Michael L Frigge, Arnaldur Gylfason, Pasha Zusmanovich, Gudmar Thorleifsson, Pall I Olason, Andres Ingason, Stacy Steinberg, Thorunn Rafnar, Patrick Sulem, Magali Mouy, Frosti Jonsson, Unnur Thorsteinsdottir, Daniel F Gudbjartsson, Hreinn Stefansson, and Kari Stefansson. Detection of sharing by descent, long-range phasing and haplotype imputation. *Nature genetics*, 40(9):1068–1075, September 2008.

[8] N Li and M Stephens. Modeling linkage disequilibrium and identifying recombination hotspots using single-nucleotide polymorphism data. *Genetics*, 165(4):2213–2233, December 2003.

[9] Yun Li, Cristen J Willer, Jun Ding, Paul Scheet, and Gonçalo R Abecasis. MaCH: using sequence and genotype data to estimate haplotypes and unobserved genotypes. *Genetic epidemiology*, 34(8):816–834, December 2010.

[10] Shin Lin, David J Cutler, Michael E Zwick, and Aravinda Chakravarti. Haplotype inference in random population samples. *American journal of human genetics*, 71(5):1129–1137, November 2002.

[11] Jonathan Marchini, D Cutler, N Patterson, M Stephens, E Eskin, E Halperin, S Lin, Z Qin, H Munro, G Abecasis, and P Donnelly. A comparison of phasing algorithms for trios and unrelated individuals. *American journal of human genetics*, 78(3):437–450, March 2006.

[12] Kimmo Palin, Harry Campbell, Alan F Wright, James F Wilson, and Richard Durbin. Identity-by-descent-based phasing and imputation in founder populations using graphical models. *Genetic epidemiology*, 35(8):853–860, December 2011.

[13] Lawrence R Rabiner. A tutorial on Hidden Markov Models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[14] P Scheet and M Stephens. A fast and flexible statistical model for large-scale population genotype data: applications to inferring missing genotypes and haplotypic phase. *American journal of human genetics*, 78(4):629–644, April 2006.

[15] M Stephens, N Smith, and P Donnelly. A new statistical method for haplotype reconstruction from population data. *American journal of human genetics*, 68(4):978–989, April 2001.

[16] Masao Ueki and Heather J Cordell. Improved Statistics for Genome-Wide Interaction Analysis. *PLoS genetics*, 8(4):e1002625, April 2012.

Nature Methods: doi:10.1038/nmeth.2307