Haplotype estimation for biobank-scale data sets

Jared O’Connell1,2,7, Kevin Sharp2,7, Nick Shrine3, Louise Wain3, Ian Hall4, Martin Tobin3, Jean-Francois Zagury5, Olivier Delaneau6,8 & Jonathan Marchini1,2,8

The UK Biobank (UKB) has recently released genotypes on 152,328 individuals together with extensive phenotypic and lifestyle information. We present a new phasing method, SHAPEIT3, that can handle such biobank-scale data sets and results in switch error rates as low as ~0.3%. The method exhibits \(O(N \log N)\) scaling with sample size \(N\), enabling fast and accurate phasing of even larger cohorts.

Estimation of haplotypes from genotypes, known as phasing, is a central part of the pipeline of many modern genetic analyses. Estimated haplotypes are important for many population genetics analyses1,2 but also form a central part of imputation algorithms that are routinely used in genome-wide association studies (GWAS)3,4. The ability to phase large data sets is especially important in the context of biobanks that comprise hundreds of thousands of genotyped samples. For example, in May 2015, the UKB released genotypes from ~152,000 samples, and this number will rise to ~500,000 in 2016. Other biobanks have already collected large-scale genetic data typed samples. For example, in May 2015, the UKB released genotypes on ~152,000 individuals together with extensive phenotypic and lifestyle information. We present a new phasing method, SHAPEIT3, that can handle such biobank-scale data sets and results in switch error rates as low as ~0.3%. The method exhibits \(O(N \log N)\) scaling with sample size \(N\), enabling fast and accurate phasing of even larger cohorts.

To describe the context in which we have developed our method, we use the following notation: \(G_i\) denotes a vector of genotypes for the \(i\)th term of \(N\) unphased individuals at \(L\) markers; \(H\) denotes a set of estimated haplotypes from other individuals at the same set of markers; and \(H^*\) is a subset of \(H\) of size \(K\). SHAPEIT2 estimates haplotypes for an individual iteratively. In each iteration, compatible haplotypes underlying \(G_i\) are sampled from a hidden Markov model (HMM) in which they are modeled as an imperfect mosaic of the haplotypes in \(H^*\). The computational complexity of this step, using the standard forward–backward algorithm for HMMs, is \(O(LK^2)\) (ref. 14). The complexity is quadratic because the model permits the haplotypes that give rise to genotypes at consecutive sites to switch for any pair of haplotypes in \(H^*\). The probability of all such transitions must be computed. One approach to ameliorating this complexity is to ensure that \(K < 2N\). The \(K\) haplotypes can be chosen randomly15 or by similarity to previous haplotype estimates for individual \(i\) (ref. 16). Alternatively, a compressed representation of \(H\) can be used by considering a region to be phased in small, disjoint windows within which only a few of the haplotypes in \(H\) are distinct12,17.

SHAPEIT2 introduced a new strategy that splits \(G_i\) into small segments of distinct haplotypes that are consistent with \(G_i\) and results in the HMM component of the method having complexity \(O(LK)\). HAPI-UR adopts a similar strategy but in such a way that scaling with \(N\) depends on the diversity of the data set and is typically superlinear. SHAPEIT2 also gains accuracy by combining linear complexity with the haplotype selection approach of IMPUTE2 (ref. 16) instead of using a compressed representation of \(H\). In this approach, in each iteration, a search is carried out to find a good conditioning set of haplotypes, \(H^*\), that can be used to update each individual. This method can be thought of as a generalization of the LRP11 approach, in which a search is carried out for just two other samples that can be used as surrogate parents when phasing each individual. These selection approaches used by SHAPEIT2 and LRP involve comparison to all haplotype pairs, which has complexity of \(O(N^2)\). For \(N > 10,000\), this complexity begins to dominate.

---

1Wellcome Trust Centre for Human Genetics, University of Oxford, Oxford, UK. 2Department of Statistics, University of Oxford, Oxford, UK. 3Department of Health Sciences, University of Leicester, Leicester, UK. 4School of Medicine, University of Nottingham, Nottingham, UK. 5Chaire de Bioinformatique, Laboratoire Génomique, Bioinformatique et Applications (Equipe d’Accueil 4627), Conservatoire National des Arts et Métiers, Paris, France. 6Département de Génétique et Développement, University of Geneva, Geneva, Switzerland. 7These authors contributed equally to this work. 8These authors jointly supervised this work.

Received 12 November 2015; accepted 9 May 2016; published online 6 June 2016; doi:10.1038/ng.3583
SHAPEIT3 enhances SHAPEIT2 in two ways that enable it to deal with very large data sets, such as the UKB study. The first advance is based on the intuition that larger sample sizes are likely to result in increased local similarity between groups of haplotypes due to the higher probability of more recent shared ancestry. This idea is exploited by using a recursive clustering algorithm to partition the haplotypes into clusters of similar haplotypes of specified size $M << 2N$. Distances are computed only between haplotypes within a cluster. This reduces the complexity of this step to $O(M^2)$, such that the complexity of the whole algorithm is dominated by the $O(N \log N)$ scaling of the clustering routine (Online Methods). We set $M = 4,000$ for the experiments in this paper, and this value leads to a reasonable tradeoff between run time and accuracy (Supplementary Fig. 1).

Our clustering method is similar to locality-sensitive hashing, and further advances in SHAPEIT development are likely by pursuing this approach further.18

The second advance involves changes to the Markov chain Monte Carlo (MCMC) sampling routine that result in additional gains in speed. As sample size grows, it becomes more likely that two individuals will have a long stretch of sequence in common within a particular window. We modified the algorithm to detect when this occurs as the algorithm proceeds. When a Hamming distance equal to zero is found between a haplotype and at least one of its conditioning haplotypes, we do not perform HMM calculations within that window from that point on. In this way, the method adapts to local patterns of haplotype sharing, and computation becomes increasingly focused on the challenging parts of the data set. On the UK Biobank Lung Exome Variant Evaluation (BiLEVE) data set of ~50,000 samples, we observed that almost 20% of 2-Mb windows will have Hamming distances of zero as the algorithm converges (Supplementary Fig. 2). We have also updated the MCMC algorithm so that threading is now performed per window rather than per individual as was done in SHAPEIT2. That is, each iteration involves updating $N \times W$ haplotype windows (typically around 2 Mb in size), and we now process each window in a separate thread as opposed to processing entire individuals on a single thread (Online Methods).

### RESULTS

#### Phasing the UK BiLEVE data set

We compared SHAPEIT3 to SHAPEIT2 (r768) and HAPI-UR (v1.01) using a large, biobank-scale data set of $N = 49,458$ individuals from the UK BiLEVE study, each of whom were genotyped on an Affymetrix Biobank genotype microarray. We phased chromosome 20, which had 15,795 SNPs after quality control filtering, for a range of sample sizes ($N = 1,000, 2,000, 5,000, 10,000, 20,000$ and $49,074$) and evaluated computational time and accuracy. We developed a new method to leverage close relatives within the data set to assess phasing accuracy. We identified 384 likely sibling pairs and obtained partially phased, accurate haplotypes for each pair by detecting IBD1 segments (regions in which siblings inherit exactly one chromosome from the same parent) using an HMM (Online Methods). After filtering, these individuals had an average of 28.7% of their heterozygous sites phased and a total of 337,634 phased heterozygous sites on which to calculate the switch error rate. One member of each sibling pair was removed from the test data set and used to provide ‘truth’ haplotypes for the other sibling. The variation in computational time and switch error rate with sample size for each method is shown in Figure 1.

As expected, accuracy and computational time increased with sample size for all methods. Although SHAPEIT2 was consistently the most accurate method for $N \geq 20,000$, its quadratic distance calculations make it more computationally challenging for larger sample sizes (hence, it was not run for $N > 20,000$). For the largest sample size ($N = 49,074$), SHAPEIT3 had a switch error rate of 1.60% and took 121 h to run (32 h when using four threads). This compared favorably to the HAPI-UR 3× (1×) method, which had a switch error rate of 2.06% (2.24%) and took 250 h (83 h) to run. Notably, SHAPEIT3 was consistently more accurate than both HAPI-UR 1× and HAPI-UR 3× and was also faster when using four threads.

RAM usage for the different methods is reported in Supplementary Table 1. We estimated that phasing chromosome 20 in all 500,000 UKB samples would require ~240 GB of RAM, which is realistic for modern systems. On larger chromosomes, a ligation strategy can be used.

#### Phasing the UK Biobank data set

As a further validation, we used SHAPEIT3 to phase the first release of the UKB data set, consisting of 152,256 individuals genotyped on a combination of the UK BiLEVE array and the UKB Axiom array (see URLs). The self-reported ancestry of these individuals is primarily white British (90.5%), with the remaining 9.5% comprising various ancestry groups. To evaluate performance, we used 72 mother–father–child trios that were detected in this data set. We used these trios to obtain a ground truth set of haplotypes. We removed the trio parents from the data set and phased the whole of chromosome 20 (16,265 genotyped sites) for the remaining 152,112 individuals. This run resulted in a median switch error rate of 0.4% and took 38.5 h using ten threads (Table 1). Supplementary Figures 3–5 show visually the accuracy for each trio child and demonstrate how this error rate corresponds to many long stretches of accurately phased sequence. When SHAPEIT3 was run on the full set of 152,112 samples, it could be seen that many samples had just a handful of switch errors per chromosome. We found that 68.5% of the inferred haplotypes consisted of correctly inferred chunks of 10 Mb or greater in length (Supplementary Table 2). By increasing the number of conditioning states and the cluster size parameter, we obtained switch error rates as low as 0.3%.

To assess the advantages of phasing such a large data set, we also ran SHAPEIT3 and SHAPEIT2 on a subset of 10,072 samples that included the trio children and obtained mean switch error rates of 1.3% and 1.1%, respectively. These runs took 2.5 h and 3.3 h using ten threads. On a subset of 1,072 samples, SHAPEIT3 had a switch error rate of 2.6% and took 0.25 h to run. We also ran SHAPEIT3 on the subset of 10,072 samples without using the new clustering routine,

---

**Table 1** Comparison of methods on the UK Biobank data set

| Sample size | Method | Clustering$^a$ | New MCMC$^b$ | Switch error rate (%) | Run time (h) | Run time scaling$^c$ | Sample size scaling |
|-------------|--------|---------------|-------------|-----------------------|-------------|---------------------|---------------------|
| 1,072       | SHAPEIT3 | No            | Yes         | 2.6                   | 0.25        | 1                   | 1                   |
|             | SHAPEIT2 | No            | No          | 1.1                   | 4.2         | 16.8                | 9.4                 |
| 10,072      | SHAPEIT3 | No            | Yes         | 1.1                   | 3.3         | 13.2                | 9.4                 |
| 10,072      | SHAPEIT3 | Yes           | Yes         | 1.3                   | 2.5         | 10.0                | 9.4                 |
| 152,112     | SHAPEIT3 | Yes           | Yes         | 0.4                   | 38.5        | 154                 | 142                 |

Each row shows performance on a subset of the full data set. Performance is measured by median switch error rate for trio children. Ten threads were used for all runs. $a$Indicates whether the new method for choosing copying states was used. $b$Indicates whether the new MCMC routine, which uses completely parallel updates and local algorithm termination, was used. $c$Relative run time as compared to the SHAPEIT3 run on a sample size of 1,072.

---
Switch errors that occur megabases apart (Supplementary Fig. 3) validated for using SHAPEIT3 to phase the UKB data set in early 2015. SHAPEIT3 was run with cluster size $M = 4,000$, which substantially improves computational complexity and, hence, run time in comparison to SHAPEIT2. Both SHAPEIT2 and SHAPEIT3 runs use $K = 100$ conditioning states.

DISCUSSION

Overall, we have demonstrated that SHAPEIT3 provides a highly accurate and scalable solution to phasing biobank-scale data sets. The extremely low switch error rates that we have obtained represent strong validation for using SHAPEIT3 to phase the UKB data set in early 2015. Switch errors that occur megabases apart (Supplementary Fig. 3) will have negligible impact on subsequent downstream imputation. Because of the multi-ancestry nature of the UK population, many thousands of UKB samples will not have European ancestry (~10% of samples in the first release). Although many new loci will be uncovered using predominantly European samples, these non-European samples will be poorly phased (and imputed) using LRP approaches that search for identity-by-descent (IBD) matches of a specific length. Such non-European samples are likely to be invaluable when deciphering the effects of new associations across ancestry groups. Given that SHAPEIT2 was demonstrated to perform well (relative to other methods) on cohorts with heterogeneous ancestry, we expect SHAPEIT3 to have similar performance. Hence, SHAPEIT3 represents a scalable method that can adapt to the multi-ancestry nature of large cohorts because of its custom selection of template haplotypes on a per-subject basis.

Reducing switch error rates to the levels produced by SHAPEIT3 can result in downstream imputation performance at low-frequency SNPs. Phasing samples altogether also avoids having to phase in batches, which would be more likely to introduce artifacts to any downstream analysis. In addition, SHAPEIT3 has also been used successfully to rephase the first release of the HaploTyper Reference Consortium (HRC) data set (see URLs), which consists of genotypes at ~40 million SNPs called from low-coverage sequencing of ~33,000 samples. The boost in imputation performance, due directly to this rephasing, has in effect already led to a boost in power to detect associations at low-frequency variants, as a large number of samples have already been imputed using HRC. Phasing sequencing-derived genotypes in this many samples could not have been performed using other LRP methods.

In other work, we have shown that large and accurately phased haplotype reference panels can be used to help phase single sequenced samples. This approach uses rare variants shared by the sequenced sample and the reference panel to efficiently select a set of template haplotypes to use in the HMM. The Oxford Statistics Phasing Server allows users to phase their samples against the HRC haplotype panel (see URLs).

Extending this approach further, we suggest that an accurately long-range-phased and imputed version of the UKB data set, at a union of all SNPs on commonly used genome-wide SNP microarrays, could act as a highly accurate reference panel for phasing newly genotyped samples with predominantly European ancestry. Combining such a panel with the multi-ancestry reference panel planned for the next release of the HRC and with the planned 100,000 Genomes Project reference panel (see URLs) would likely provide a reference panel useful for phasing samples with a wide range of ancestries.

METHODS

Methods and any associated references are available in the online version of the paper.

Note. Any Supplementary Information and Source Data files are available in the online version of the paper.
J.M. carried out the experiments. N.S., L.W., I.H. and M.T. provided the UK BiLEVE data set. J.M. and O.D. supervised the research. J.O'C., K.S., J.-F. Z., O.D. and J.M. wrote the manuscript.

COMPETING FINANCIAL INTERESTS
The authors declare no competing financial interests.

Reprints and permissions information is available online at http://www.nature.com/reprints/index.html.

1. Hellenthal, G. et al. A genetic atlas of human admixture history. Science 343, 747–751 (2014).
2. 1000 Genomes Project Consortium. A global reference for human genetic variation. Nature 526, 68–74 (2015).
3. Marchini, J. & Howie, B. Genotype imputation for genome-wide association studies. Nat. Rev. Genet. 11, 499–511 (2010).
4. Howie, B., Fuchsberger, C., Stephens, M., Marchini, J. & Abecasis, G.R. Fast and accurate genotype imputation in genome-wide association studies through pre-phasing. Nat. Genet. 44, 955–959 (2012).
5. Hoffmann, T.J. et al. Imputation of the rare HOXB13 G84E mutation and cancer risk in a large population-based cohort. PLoS Genet. 11, e1004930 (2015).
6. Chen, Z. et al. China Kadoorie Biobank of 0.5 million people: survey methods, baseline characteristics and long-term follow-up. Int. J. Epidemiol. 40, 1652–1666 (2011).
7. Wann, L.V. et al. Novel insights into the genetics of smoking behaviour, lung function, and chronic obstructive pulmonary disease (UK BiLEVE): a genetic association study in UK Biobank. Lancet Respir. Med. 3, 769–781 (2015).
8. Delaneau, O., Zagury, J.-F. & Marchini, J. Improved whole-chromosome phasing for disease and population genetic studies. Nat. Methods 10, 5–6 (2013).
9. O’Connell, J. et al. A general approach for haplotype phasing across the full spectrum of relatedness. PLoS Genet. 10, e1004234 (2014).
10. Martin, H.C. et al. Multicohort analysis of the maternal age effect on recombination. Nat. Commun. 6, 7846 (2015).
11. Kong, A. et al. Detection of sharing by descent, long-range phasing and haplotype imputation. Nat. Genet. 40, 1068–1075 (2008).
12. Williams, A.L., Patterson, N., Glessner, J., Hakonarson, H. & Reich, D. Phasing of many thousands of genotyped samples. Am. J. Hum. Genet. 91, 238–251 (2012).
13. Loh, P.-R., Palamara, P.F. & Price, A.L. Fast and accurate long-range phasing and imputation in a UK Biobank cohort. Nat. Genet. http://dx.doi.org/10.1038/ng.3571 (2016).
14. Scheet, P. & Stephens, M. A fast and flexible statistical model for large-scale population genotype data: applications to inferring missing genotypes and haplotypic phase. Am. J. Hum. Genet. 78, 629–644 (2006).
15. Li, Y., Willer, C.J., Ding, J., Scheet, P. & Abecasis, G.R. MaCH: using sequence and genotype data to estimate haplotypes and unobserved genotypes. Genet. Epidemiol. 34, 816–834 (2010).
16. Howie, B.N., Donnelly, P. & Marchini, J. A flexible and accurate genotype imputation method for the next generation of genome-wide association studies. PLoS Genet. 5, e1000529 (2009).
17. Delaneau, O., Marchini, J. & Zagury, J.-F. A linear complexity phasing method for thousands of genomes. Nat. Methods 9, 179–181 (2012).
18. Koga, H., Ishibashi, T. & Watanabe, T. Fast agglomerative hierarchical clustering algorithm using locality-sensitive hashing. Knowl. Inf. Syst. 12, 25–53 (2007).
19. McCarthy, S. et al. A reference panel of 64,976 haplotypes for genotype imputation. BioRxiv http://dx.doi.org/10.1101/0353170 (2015).
20. Sharp, K., Kretzschmar, W., Delaneau, O. & Marchini, J. Phasing for medical sequencing using rare variants and large haplotype reference panels. Bioinformatics http://dx.doi.org/10.1093/bioinformatics/btw065 (2016).
ONLINE METHODS

A subquadratic method for haplotype distance calculations. At the ith iteration of SHAPEIT2, a pair of haplotypes (H(i), H(i+1)) for the ith individual of N total individuals is resampled from an HMM. The hidden states of this HMM are a set of K conditioning haplotypes, H∗, chosen from the remaining 2(N−1) estimated haplotypes, H, in the cohort. An optimal choice for H∗ would be the set of conditioning haplotypes that would give rise to (H(i), H(i+1)) with the highest probability, but this set is unknown. As a proxy, we choose haplotypes that are locally similar. The region to be phased is divided into windows of a specified size. In each window, H∗ is formed from the K haplotypes in H that are closest in Hamming distance to the haplotype estimates (H(i−1), H(i+1)) from the previous iteration. The intuition is that, in small windows, closeness in Hamming distance is correlated with shared ancestry. In each window, all 2N haplotypes are compared to one another. Consequently, the distance calculation of SHAPEIT2 has complexity of O(N²). When sample sizes become large, say N > 10,000, this complexity starts to dominate the computational cost of the algorithm.

In SHAPEIT3, we circumvent the O(N²) complexity for distance comparisons by a two-step procedure:

1. Cluster the 2N haplotypes into clusters of size M.
2. For each individual, form the set H∗ from the other M−1 haplotypes belonging to the same cluster. An individual’s haplotypes may be in two distinct clusters, in which case there are 2(M−1) haplotypes to consider.

Step 2 requires distance computations between the M haplotypes within a cluster and so has complexity O(M²), independent of N. We now describe a simple clustering method, divisive k-means clustering, to perform step 1. We show that this method has complexity O(NlogN). Consequently, our new algorithm exhibits overall complexity of only O(NlogN).

Divisive k-means clustering to partition the data. k-means clustering is a well-known technique for uniquely assigning samples to one of k clusters on the basis of similarity for some real-valued attribute. Assignments are performed so as to minimize the within-cluster sum of squares

\[ \sum_{j}^{2N} \left| H_j - \mu_c \right|^2 \]

where \( \mu_c \) is the cluster mean for class c and \( C_j \) denotes the cluster membership of haplotype j. This objective function can be minimized by iteratively assigning class labels to each observation on the basis of which cluster mean is closest and then recalculating the cluster means. When the distance measure is Euclidean, convergence is guaranteed. We implemented this routine, allowing up to ten iterations (or stopping when class labels stabilized). Ten iterations are often not sufficient for convergence, but we only require a rough partitioning. We used k-means clustering with \( k = 2 \) to partition clusters of haplotypes recursively. Each cluster is split into two smaller clusters. The recursion is terminated when cluster size is \( < M \). Upon termination, we ‘top up’ the cluster at the leaf of the bifurcating tree with haplotypes from the closest leaf. This last step ensures that each haplotype is compared to \( M - 1 \) other haplotypes.

Formally, we wish to find a vector of labels \( C = \{C_1, C_2, ..., C_{2N}\} \) where \( C_i \) tells us the cluster to which haplotype \( H_i \) belongs. We refer to \( C_i \) as the primary cluster of \( H_i \). We also store a dictionary \( D \) of sets, which we call secondary clusters. Every set in \( D \) has exactly \( M < 2N \) elements. It is possible for a haplotype to be in more than one secondary cluster, but haplotypes will only ever have one primary cluster (determined by \( C_i \)). We assume that we have the following functions available: k-means (H) and Eucl(H, \( \mu_c \)). The first performs the k-means clustering routine with \( k = 2 \) on H and returns a vector of class labels, and the second calculates the Euclidean distance between \( H_i \) and \( \mu_c \). Our algorithm is described using pseudocode in the Supplementary Note.

In each iteration of the SHAPEIT3 algorithm, we find the C primary cluster labels and build the dictionary \( D \) of secondary clusters. For each individual i, we choose K conditioning haplotypes from \( H^* = D[C_{i-1}] \cup D[C_i] \) on the basis of minimum Hamming distances but exclude individuals estimated to be IBD2 within the region. We set the cluster size to \( M = 4,000 \) for all the experiments described in this paper. We note that the final set of K conditioning haplotypes is chosen on the basis of minimum Hamming distance, whereas the k-means routine uses Euclidean distance. Hamming distance is appropriate for pairs of binary vectors; we use this where possible because such distances can be calculated very rapidly via lookup tables. The k-means routine needs to use a distance measure appropriate for real numbers, as the means of the clusters are unlikely to be 0 or 1 but rather something in the interval [0,1]; hence, Euclidean distance was used. We suspect that this part of the algorithm can be improved via careful choice of the SNPs used to calculate Euclidean distances between haplotypes, for example, by preferentially focusing on rare SNPs.

We further decrease the computational cost of our routine by thinning the SNPs on which the k-means analysis is performed. For each k-means clustering performed (each leaf of the recursion), Euclidean distances are only calculated on SNPs \( i \sim [8(k + \alpha); k \in 0 \sim L/8] \), where \( \alpha \in [0,1,2,3,4,5,6,7] \), and are sampled randomly within each leaf. In words, we only use every eighth SNP (starting from a random offset) in our clustering routine. Thus, in practice, 4NL/8 differences (2 centroids, 2N haplotypes and L/8 SNPs) are calculated per iteration of k-means clustering; when clustering N samples, we include this description for completeness, but we remove the ‘8’ from the denominator henceforth for clarity. The initial iteration of k-means clustering requires 4NL calculations to calculate the distance between the N haplotypes and two clusters for L SNPs. As a rough approximation, we assume that k-means clustering divides the data into two clusters of equal size. Consequently, after the dth recursion, we have \( 2^d \) clusters of size \( N/2^d \). The recursion continues until a cluster size \( < M \) is reached. This will take \( \log_2 (N/M) \) recursions. Hence, we can derive the computational complexity of the clustering routine quite simply as

\[ \text{complexity} = \sum_{d=0}^{\log_2 N/M} 2^d \cdot 4NL/2^d = 4NL \log_2 N/M \]

Because \( L \) and \( M \) are constants, our clustering routine has complexity of \( O(N\log N) \).

Modified MCMC sampling routine. We implement two modifications to the MCMC sampling routine of SHAPEIT2. First, when a Hamming distance of zero is found between a haplotype and at least one conditioning haplotype, we do not perform HMM calculations within that window in the current iteration. In such cases, it is unlikely that a different haplotype will be generated in preference to this perfect match. In other words, when there is evidence locally that the algorithm has converged, we stop updating that individual. We refer to this as an ‘adaptive algorithm termination’. Therefore, we simply carry forward the haplotype generated by the previous iteration. As shown in Supplementary Figure 2, this can be expected to happen quite frequently as \( N \) increases, leading to considerable savings in computational time.

The second modification is an approximation to the MCMC sampling routine that offers substantial additional gains in speed when using multiple threads. In SHAPEIT2, the parallelization scheme updates the phase of T samples in parallel using T threads and conditioning on the \( N − T \) other samples already processed. The problem with this approach is that a given thread becomes idle as soon as it has finished processing its assigned sample before the other threads and therefore has to wait for the other threads to finish. Because of this synchronization that has to be repeated many times, the parallelization efficiency of this scheme decreases with the number of threads. In SHAPEIT3, we introduce a new parallelization scheme based on another approximation of the MCMC process. This scheme takes advantage of the fact that SHAPEIT updates the phase of samples within windows of usually a few megabases in size: it splits the N samples into W overlapping windows before any MCMC iteration and distributes the \( N \times W \) jobs to be done into T distinct queues, one for each thread. Then, a thread performs the jobs for the window–sample pairs in its queue independently of the other queues and therefore does not need any particular synchronization, resulting in better usage of all CPU cores. Once all the jobs are completed, the haplotype for each sample is updated and we move on to the next MCMC iteration. Of note, this scheme implicitly assumes an MCMC scheme in which the phase of all samples is updated in parallel at the same time. In other words, we move from one Markov chain state to the next by updating all samples simultaneously; this differs from the standard MCMC scheme in which we move from one
state to the next by updating only one sample. We refer to the new method as ‘completely parallel updating’. We assessed the performance of this approach by running SHAPEIT3 without the new clustering routine on the subset of the UKB data set with 10,072 samples and compared it to the performance of the SHAPEIT2 run on the same data set. The median switch error rate was 1.1% for both methods, indicating little difference in performance when using this new MCMC routine (Table 1). Performance for increasing numbers of threads is shown in Supplementary Table 4.

Creating a validation data set to assess accuracy. SHAPEIT3 was initially developed and tested on the UK BiLEVE data set, which contains a large number of apparent sibling pairs but no identifiable mother–father–child trios. Regions in which siblings inherit exactly one chromosome from the same parent (IBD1 regions) can be phased using simple Mendelian rules. Thus, we developed a new method for inferring the IBD status of siblings with the aim of constructing a set of ‘known’ haplotypes against which we could compare the accuracy of the different methods.

We identified 384 pairs of individuals with a kinship coefficient >0.35, indicating that they are likely first-degree relatives (sibling pairs or parent–child duos). Parents will share 100% of chromosomes IBD1 with their children, and the genomes of siblings will (on average) be 25% IBD0, 50% IBD1 and 25% IBD2. No pairs of individuals had close to 0% IBD0, suggesting that these samples were sibling pairs, not parent–child duos22. We can accurately phase loci in detected IBD1 regions using simple Mendelian rules. This will generate accurate haplotypes for validation purposes, albeit partially phased ones.

We infer regions of IBD1 sharing by a two-stage approach. First, we apply a simple HMM to the genotypes for each sibling pair (Supplementary Note), with three unobserved states indicating IBD status. However, because this model does not account for linkage disequilibrium (IBD sharing from more distant ancestors), it is vulnerable to inferring spurious stretches of IBD1. This issue has been noted in previous work on phasing siblings with data missing for the parents22. We circumvent it via simple post-hoc filtering of short (<10-cM) IBD1 and IBD2 segments. Supplementary Figure 6 shows the consequences of this filtering step on IBD state for a pair of probable siblings across chromosome 20. The filtering step did indeed reduce the switch error rate for all methods (Supplementary Fig. 7) and also flattened the distribution of switch error rates with respect to the percentage of phase-resolved sites for the corresponding individual (Supplementary Fig. 8).

We note that the switch error rates obtained when using these validation data were higher than those for the trio phased validation haplotypes created in the UKB experiments. This suggests that our sibling pair validation haplotypes were not as accurate as those produced via trio phasing. We can think of two likely causes. We filter out short stretches of (likely spurious) IBD1, but such spurious segments may flank longer (true) IBD1 segments, introducing some errors on the edges of segments. Second, genotyping errors will result in incorrect phasing, and there is greater power to detect such errors in the trio setting. Nevertheless, this validation data set provides a reasonable means for the relative comparison of methods and was of great usefulness when developing SHAPEIT3.

Methods comparison using the UK BiLEVE data set. For the results presented in Figure 1, which compares methods, each phasing run was performed on independent Amazon EC2 m2.2xlarge instances to avoid any possibility of diminished performance from other running processes. The elapsed time of each run was measured with the GNU time command. HAPI-UR results were generated with v1.01 of the software, using a window size of 80 SNPs as suggested in the HAPI-UR manual for a microarray of the density used. SHAPEIT2 results were generated with version r778 of the software using default parameters.

21. Lloyd, S. Least squares quantization in PCM. IEEE Trans. Inf. Theory 28, 129–137 (1982).
22. Manichaikul, A. et al. Robust relationship inference in genome-wide association studies. Bioinformatics 26, 2867–2873 (2010).
23. Hinch, A.G. et al. The landscape of recombination in African Americans. Nature 476, 170–175 (2011).