Evaluation of Vicinity-based Hidden Markov Models for Genotype Imputation

Su Wang¹, Miran Kim², Xiaoqian Jiang³, Arif Harmanci¹

1 Center for Precision Health, School of Biomedical Informatics, University of Texas Health Science Center, Houston, TX, 77030.

2 Department of Computer Science and Engineering and Graduate School of Artificial Intelligence, Ulsan National Institute of Science and Technology, Ulsan, 44919, Republic of Korea.

3 Center for Secure Artificial intelligence For hEalthcare (SAFE), School of Biomedical Informatics, University of Texas Health Science Center, Houston, TX, 77030, USA.

Abstract

The decreasing cost of DNA sequencing has led to a great increase in our knowledge about genetic variation. While population-scale projects bring important insight into genotype-phenotype relationships, the cost of performing whole-genome sequencing on large samples is still prohibitive. In-silico genotype imputation coupled with genotyping-by-arrays is a cost-effective and accurate alternative for genotyping of common and uncommon variants. Imputation methods compare the genotypes of the typed variants with the large population-specific reference panels and estimate the genotypes of untyped variants by making use of the linkage disequilibrium patterns. Most accurate imputation methods are based on the Li-Stephens hidden Markov model, HMM, that treats the sequence of each chromosome as a mosaic of the haplotypes from the reference panel. Here we assess the accuracy of local-HMMs, where each untyped variant is imputed using the typed variants in a small window around itself (as small as 1 centimorgan). Locality-based imputation is used recently by machine learning-based genotype imputation approaches. We assess how the parameters of the local-HMMs impact the imputation accuracy in a comprehensive set of benchmarks and show that local-HMMs can accurately impute common and uncommon variants and can be relaxed to impute rare variants as well. The source code for the local HMM implementations is publicly available at <https://github.com/harmancilab/LoHaMMer>.

Introduction

As the cost of DNA sequencing is decreasing, the number of available genome sequences is increasing at a fast pace¹⁻⁴. DNA sequencing is also the fundamental step for technologies such as RNA sequencing and ChIP-Sequencing⁵. Currently, there are millions of genomic sequences available and many more are expected⁶⁻⁸. As the genomic data is used more prevalently in the clinic and in translational research^{9,10}, the genetic data size is available in many different scenarios, even including the citizen scientists from the general population¹¹. Genetic data is deposited widespread in many places (including personal computers and even phones) and it made its way well into the fields of recreational genetics¹². This is made possible by extensive mapping of the genetic differences between populations and efficient methods that can sift through massive databases for searching for relatives¹³. These are made possible by population-scale projects such as UKBiobank¹⁴. The third parties aim at capitalizing on these data by generating “new and interesting” datasets from rare phenotypes, diseases, and rare populations and individuals¹⁵⁻¹⁸. The

genetic data is treated as a new type of commodity through brokers^{19,20}. As the COVID19 pandemic showed, sharing and using genetic data out of the research lab and in the field and clinic will be routine tasks in many cases^{21,22}.

One of the main uses of genetic data is performing genotype-phenotype associations using genome-wide association studies (GWAS or GWA study)^{23–26}. For this, a large cohort is generated and the individuals are genotyped by sequencing. Next, the phenotype of interest (Intelligence quotient, height, body-mass index, blood glucose levels, etc.) is measured from all the individuals. Finally, the measured genotypes for all the variants are tested for association with the GWA studies, most variants are found to be in intergenic regions out of the protein-coding exons. Thus, it is necessary to perform genotyping on the whole genome scale using, for example, whole-genome sequencing (WGS) to ensure that the causal variants, not another variant that is in linkage disequilibrium (LD)^{27,28}, can be identified as accurately as possible. This, however, is not cost-effective because large samples must be whole-genome sequenced²⁹. To get around this, genotyping arrays are used for genotyping and decreasing the cost³⁰. The genotyping arrays are designed to genotype only a sparse set of variants from the genome. These variants are then input to in-silico genotype imputation algorithms^{31,32}, which impute and “fill-in” the un-genotyped (or untyped for short) variants. The main idea behind the imputation algorithms is to make use of the known haplotype structure of the whole genome and estimate the genotypes of the untyped variants using the genotypes of typed-variants that are correlated at the haplotype level³³. The haplotype structure arises because the alleles are inherited between generations by a limited number of crossing-overs at the recombination hotspots between homologous chromosomes³⁴. This causes long chunks of haplotypes to be inherited as a single unit. Although the length of conserved chunks (identity-by-descent segments³⁵) decreases as the relationship distance increases, it can still be detected even with 20-25 generations of separation between individuals^{36,37}. The imputation algorithms focus on making use of conserved chunks of haplotypes (i.e., frequent haplotypes) that are shared among unrelated individuals in the population. Imputation methods are also used for imputing variants identified by the RNA sequencing and whole-exome sequencing and for fine mapping of the variants from association studies.

The current state-of-the-art imputation methods such as BEAGLE, Minimac, and IMPUTE suite make use of the hidden Markov model (HMM)^{38,39} based approach that is developed by Li and Stephens^{40–45}. HMM treats each haplotype as a “state” and analyzes the probabilities of all the “paths” that pass through the states to generate the alleles that are typed by the array⁴¹. This way, HMM-based methods can assign probabilities to the imputed genotypes using the probabilistic model imposed by the Li-Stephens haplotype model. The HMM takes the typed variants and the reference panel as input and imputes all the variants that exist on the reference panel but are untyped by the genotyping array. While HMM models provide good accuracy of imputation, they may fail at imputation of rare variants as these variants are represented on at least as rare haplotypes⁴⁶. However, as the size of reference panels increases, the rare variants can be more accurately predicted⁴⁷.

Here, we focus on Li-Stephens HMM-based imputation models and assess the performance of “local-HMMs”, i.e., the HMM evaluates the paths over only a short stretch of variants around the untyped variants. While several methods have tested different parametrizations of the state-of-the-art methods, we implemented the local-HMM methods to have full control over how the parameters impact the imputation. The locality-based approaches have been used in different scenarios, for instance with linear imputation models and with Deep Learning-based imputation models^{48–50}, where the imputation is performed on the typed variants that are in the vicinity of the untyped variant. Also, IMPUTE and BEAGLE

make use of a sliding-window, as long as 40 centimorgans (cM) to cut corners in computation. This parameter was not extensively assessed in terms of its impact on imputation accuracy. We implemented the per-position posterior probability estimation (we refer to this as the MAP method) by the forward-backward algorithm. We also implement the inference of the maximum-likelihood HMM path (referred to as the ML method), which represents the most likely mosaic of reference haplotypes that gives rise to the genotypes of the typed variants. On these methods, we analyze the size of the window, positioning of the target within the window, number of tag variants on the window, and the effective population size. It should be noted that we focus on the phased genotype imputation problem, i.e., we assume that the genotypes are phased. This is a reasonable assumption since pre-phasing has been showed to improve the time complexity of the imputation method substantially while conferring a very small performance penalty⁴⁴.

One of the main advantages of the locality-based approaches is that they can be constrained in terms of computational requirements without the need to load the whole genome into memory or running the HMM inference methods on whole chromosomes. This way, the architecture of imputation algorithms on a cloud can be structured accordingly, for example, by using different models in different parts of the genome. On another front, the recent development of privacy-aware genotype imputation methods make use of the vicinity-based models and therefore our study can inform these methods about the locality parameters that must be considered and evaluated for increasing the imputation accuracy while providing privacy and confidentiality to the genetic data. The implementation of the local-HMMs, named LoHaMMer, is publicly available to download from <https://github.com/harmancilab/LoHaMMer>.

Results

We briefly describe the HMM-based imputation techniques, the parameters, and the evaluation approach. We finally present the imputation accuracy evaluations.

Overview of the Vicinity-based HMMs

Genotype imputation is summarized in Figure 1. The genotype imputation process takes as input the variant phased genotypes matrix, $G_{M \times V}$, individuals. As we are evaluating the phased imputation process, G is pre-phased using a phasing algorithm such as Eagle⁵¹. $G_{i,j}$ holds the phased genotype of the j^{th} variant for the i^{th} individual, i.e., $G_{i,j}^{(h)} \in \{0,1,\emptyset\}$, $1 \leq i \leq M$, $1 \leq j \leq V$, where h indicates the paternal/maternal copy for the genotype, i.e. $h \in \{0,1\}$. $G_{i,j}^{(h)} = \emptyset$ indicates the missing genotype that will be imputed using the reference panel. We denote the set of indices of the untyped variants with $j_\emptyset = \{j \mid G_{i,j}^{(h)} = \emptyset\}$. Imputation also takes the reference genotype matrix containing $H_{N \times V}$ of N haplotypes over the same V variants that correspond to the columns of G . Similar to G , $H_{i,j} \in \{0,1\}$, $1 \leq i \leq N$, $1 \leq j \leq V$.

Li-Stephens Markov Model. Our evaluations use the Markov model defined by the standard Li-Stephens model⁴⁰, where the haplotypes of each query individual are modeled as a “mosaic of the reference haplotypes” such that pieces of reference haplotypes (consecutive variant alleles on a haplotype) are concatenated to each other. This model describes a probability distribution on possible “paths” that pass over the reference haplotypes. In this model, the transitions between the haplotypes and errors on the haplotypes are probabilistic. In the simplest sense, the minimal number of haplotype transitions and allelic errors can be thought of as the most likely path that describes the query haplotype. The basic idea is to

pin-down the tag variant alleles on the paths, and estimate the probability of alleles at the untyped variants:

$$\forall j \in j_{\emptyset}; P(G_{i,j}^{(h)} = 0) = \sum_k \left(\begin{array}{c} \text{Probability of } k^{th} \text{ path such} \\ \text{that } j^{th} \text{ variant allele is 0} \end{array} \right)$$

In this model, the haplotypes of the reference panel correspond to the states of the Li-Stephens Markov model. Each state (haplotype) emits an allele at a variant position $1 \leq j \leq V$. In addition, the transitions between the states (i.e., the switches between haplotypes) at variant j are dependent only on the genetic distance between the variants at indices j and $(j + 1)$. The genetic distance measures the probability of recombination taking place between these two variants. In the Markov model, recombination corresponds to a state-switch whereby the state (i.e., the haplotype) makes a transition to a new state. However, the recombinations occur as homologous chromosomes crossover in the course of meiosis. The rate of recombinations changes depending on the position on the genome, i.e., some parts of the genome are more likely to harbor recombinations than others. Thus, the prevalence of recombination events along the genome is quantified in terms of genetic distance that is measured in centimorgans (cM), a measure of recombination probability between two loci. Given two variants at indices $(j - 1)$ and j , the probability of recombination is modeled as:

$$P_j^{(R)} = \frac{1}{N} \times \left(1 - \exp \left(-4 \times N_e \times \frac{\Delta R_j}{N} \right) \right)$$

$$\Delta R_j = R_j - R_{j-1}$$

where $P_j^{(R)}$ denotes the probability that there is a recombination event (i.e. Markov chain stays on the same state), ΔR_j denotes the genetic distance between variants at indices $(j - 1)$ and j , and N_e denotes the effective population size. It is important to note that the probability of recombination depends only on the position of the variant and not the actual haplotype. This is widely used in HMM-based imputation methods to decrease computational costs. The probability of that a recombination does not take place can be computed from $P_j^{(R)}$:

$$P_j^{(NR)} = 1 - \left((N - 1) \times P_j^{(R)} \right)$$

where all recombination events are accounted for and removed from 1 and $P_j^{(NR)}$ indicates that there is no recombination between variants at indices $(j - 1)$ and j . From the above equation for $P_j^{(R)}$, Increasing population size implies a higher probability of recombination, i.e., larger effective population size indicates more complex recombination patterns as the probability of switching between haplotypes (or states) increases. Given the query individual's phased genotypes, $G_{i,j}^{(h)}$ and the reference haplotype data, $H_{i,j}$, the hidden Markov model is defined based on these equations using the transition and emission probabilities:

$$\tau_j(b \rightarrow a) = \begin{cases} P_j^{(NR)} & ; a = b \\ P_j^{(R)} & ; a \neq b \end{cases}$$

$$e_j \left(G_{i,j}^{(h)}, a \right) = \begin{cases} \epsilon_{allele} = 10^{-4}; & H_{a,j} \neq G_{i,j}^{(h)} \\ 1 - \epsilon_{allele} & ; H_{a,j} = G_{i,j}^{(h)} \end{cases}$$

where $\tau_j(b \rightarrow a)$ denotes the transition probability from haplotype b to a at variant index j from the previous variant at index $(j - 1)$ and $e_j \left(G_{i,j}^{(h)}, a \right)$ denotes the emission probability of the allele $G_{i,j}^{(h)}$ from the a^{th} haplotype. The emission probability depends on the alleles of the query individual; if the allele on the a^{th} haplotype matches query individual's allele, a high emission probability is assigned, otherwise allele error probability, ϵ_{allele} , is assigned as the emission probability.

Using the above equations and Li-Stephens Model, we use two approaches for inferring the haplotype states at every typed variant.

Inference of Posterior State (Haplotype) Probabilities. First approach the estimation of per-typed-variant estimate of posterior probabilities of each haplotype and assignment of the maximum *a-posteriori* (MAP) estimate of the alleles at the untyped variants. For this, we make use of the forward-backward algorithm⁵², which is a well-known dynamic programming algorithm that is used to efficiently compute the state probabilities at each step of the HMM as

$$P \left(S_{i,j}^{(h)} = a \right) = \underbrace{P \left(G_{i,[1,j]}^{(h)} | S_{i,j} = a \right)}_{Forward\ Variable} \times \underbrace{P \left(G_{i,[j+1,V]}^{(h)} | S_{i,j} = a \right)}_{Backward\ Variable}$$

where $S_{i,j}^{(h)}$ denotes the state (haplotype) of the HMM at variable index j for individual i 's parental copy h , $G_{i,[1,j]}^{(h)}$ denotes the sequence of alleles for variants in $[1, j]$ on individual i 's parental copy h , and $P \left(G_{i,[1,j]}^{(h)} | S_{i,j}^{(h)} = a \right)$ is the forward-variable and it denotes the probability of emitting the allele sequence $G_{i,[1,j]}^{(h)}$ given that HMM is at state a at the variant position j . Backward-variable is similarly defined for the rest of the allele sequence that is backward of j^{th} variant, i.e., $G_{i,[j+1,V]}^{(h)}$. The forward and backward variables are computed using efficient recursion relations (See Methods). After the posterior probability at each variant position j and for each state a is computed, we can estimate the posterior probability of each allele at each untyped position:

$$\forall j \in j_{\emptyset}, \quad P \left(G_{i,j}^{(h)} = t \right) = \sum_{a \in N} P \left(S_{i,j}^{(h)} = a | H_{a,j} = t \right); t \in \{0,1\}$$

where the untyped variant allele t 's probability is estimated by marginalizing over the states a for which the corresponding haplotype has an allele t . As we describe below, we evaluate 2 different approaches for marginalizing over the haplotypes.

Maximum-Likelihood Mosaic-Haplotype (Viterbi). Vicinity-based HMMs focus only on the vicinity of untyped variants. The main idea is to select a vicinity for imputation of a target variant.

$$\check{S}_{i,[1,V]}^{(h)} = \underset{S_{[1,V]}}{\operatorname{argmax}} \left\{ P \left(G_{i,[1,V]}^{(h)} | S_{[1,V]} \right) \right\}$$

where $P \left(S_{[1,V]} | G_{i,[1,V]}^{(h)} \right)$ denotes the likelihood of the state sequence $S_{[1,V]}$ given the allele sequence of all variants in $[1, V]$ for i^{th} individuals. $\check{S}_{i,[1,V]}^{(h)}$ denotes the maximum-likelihood (ML) state sequence for

i^{th} individual's haplotype h . This state sequence represents the most likely mosaic haplotype that gives rise to the variant alleles $G_{i,[1,V]}^{(h)} = (G_{i,1}^{(h)}, G_{i,2}^{(h)}, \dots, G_{i,V}^{(h)})$. This state sequence can be inferred using a dynamic programming algorithm, namely the Viterbi algorithm⁵³ that efficiently identifies the maximum-likelihood state sequence similar to the forward algorithm.

After the ML state sequence is computed using the Viterbi algorithm, we can assign the alleles on the haplotypes that correspond to the ML state sequence at every variant:

$$\forall j \in j_{\emptyset}, \quad \check{G}_{i,j}^{(h)} = H_{k,j}; k = \check{S}_{i,j}^{(h)}$$

where $\check{G}_{i,j}^{(h)}$ denotes the ML allele for the untyped variant j that is assigned to individual i 's haplotype h . The Viterbi algorithm does not immediately assign a score for the ML allele. We aggregate the vicinity information to assign a score for the imputed ML allele.

Locality Parameters of Imputation. We evaluate the effect of changing parameters on the accuracy of genotype imputation. The forward-backward and Viterbi-based imputation algorithms sequentially analyze the variants while keeping track of the scores and probabilities for each state. They can be performed using all of the variants on each chromosome as the LD information is confined generally to individual chromosomes and inter-chromosomal LD information, while detectable, are very weak⁵⁴. These are out-of-scope of the imputation methods that we evaluate. Using whole chromosomes in imputation enables the algorithm to integrate the linkage information from all positions on the chromosomes. On the other hand, the linkage information tends to decrease quickly while imputing an untyped variant, e.g., the identity-by-descent segment length (length of conserved haplotypes) decreases quickly among generations (25 generations separation have on average 2cM conservation³⁶). This information can be integrated into MAP and ML-based imputation by a sliding-window where the variants outside a local window are not used for imputation. This can help decrease the computational requirements. For example, BEAGLE uses a large sliding window (length 30 cMs) and merges the consecutive windows to infer the forward and backward variables. In our study, we run forward-backward and Viterbi algorithms solely on the local windows around the untyped variants and use these “local-HMMs” to impute the untyped variants. For instance, if we are using a local window of length 0.5 cMs, the ML state sequence is computed only for the local vicinity of the is assigned as:

$$\check{S}_{i,[k,l]}^{(h)} = \underset{S_{[k,l]}}{\operatorname{argmax}} \left\{ P \left(S_{[k,l]} \mid G_{i,[k,l]}^{(h)} \right) \right\}, |R_k - R_l| \approx 0.5 \text{ cM}$$

where the ML state sequence, $\check{S}_{i,[k,l]}^{(h)}$, is confined to the variant indices $[k, l]$ whose genetic distance is approximately 0.5 cMs. The forward-backward computations are similarly confined to the local windows based on genetic distance cutoffs.

We test different local window lengths and evaluate the impact of window length on the allele imputation accuracy. We utilize a sliding window with lengths from 0.1 cM upto 1 cMs and compute the imputation accuracy (See Metrics). Another important factor is the positioning of the untyped target variant within the local window. It is expected that the LD information can be integrated more accurately if the untyped variant is centered around the local window. It is, however, not clear to what extent “target-to-center distance” affects the imputation accuracy. For each untyped target variant, we first identified the typed variants that will be used for imputation that satisfies the local window length and target-to-center

distance criteria using ML and MAP approaches with the selected population size and allelic probability assignment procedure.

Evaluation Setup and Metrics. We use genotype data from the 1000 genomes Project's Phase 3⁵⁵. We focus on the variants on chromosomes 19, 20, and 22 for extensive evaluation and exclude the multi-allelic SNVs and indels. Among these data, chromosome 22 is used to evaluate different parameter combinations. To decrease computational requirements with the parameter combinations, we focused on the region chr22:25,000,000-35,000,000. In the evaluations, we randomly selected 1000 individuals as the phased reference panel and 200 individuals (with known genotypes) for estimating evaluation. After we evaluated the parameters, we selected the optimal parameter set and validated the imputation on chromosomes 19 and 20. To define the typed (tag) variants, we extracted the positions of the variants that are genotyped on the Illumina Duo 1M genotyping array platform⁵⁶. This enables us to perform evaluations on a realistic test case as the Illumina's array is used in several large-scale projects including the HAPMAP project⁵⁶. We used all the variants that map to the positions that overlap with Illumina Duo platform as the typed variants and the remaining variants are assigned as untyped variants that are imputed. After extracting the variants, we phased the genotypes using EAGLE2⁵¹. We did not rely on the phasing that is provided by 1000 Genomes because it may artificially increase the accuracy of the imputation accuracy for the local HMM approach in comparison to established methods. The phased typed variants are input to LoHaMMer with different parameters for imputation. After the untyped variants are imputed, we computed the (1) "Genotype R^2 "⁵⁷, (2) Genotype concordance (all and non-reference genotypes), and (3) Precision-Recall curves based on the imputed probabilities. We compare the implemented locality-based HMMs with BEAGLE, which is used as the baseline method for imputation. The imputation accuracy is classified among variants with respect to the range of minor allele frequency (MAF) and with respect to the chromosomal position.

Evaluation of Imputation Accuracy with Changing Locality Parameters

For assessing the imputation accuracy with changing parameters, the variants are classified into "common" (MAF>0.05) and "uncommon" (MAF<0.05) variants. We tested the impact of the 4 different parameters, ($l_w, N_e, l_{c2t}, n_{tag}$) that describe the local HMM window around the untyped (target) variants. Rather than computing all parameter combinations, we selected a range for each parameter and we evaluated the impact of one parameter while keeping others constant. We used ($l_w, N_e, l_{c2t}, n_{tag}$) = (0.3, 10⁴, 0.05, 10000) as the default parameter values.

Local Window Length (l_w). We first evaluated the impact of local window length on the imputation accuracy. We used local window lengths of $l_w \in \{0.02, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 1\}$ cM. Figures 2c and 2d show the genotype R^2 distribution and non-reference genotype concordance of the common variants on chromosome 22 with the changing local window length. The imputation accuracy of the 3 methods, MAP, ML, and BEAGLE (baseline) are reported. As expected, we observed that the accuracy increases with increasing window lengths. For window lengths above 0.3 cMs, we observed that there is less than 0.05% increase (91% vs 90.5%) in the R^2 accuracy and similarly, 0.5% increase in the non-reference genotype concordance when MAP and BEAGLE are compared. For the uncommon variants, the minimum window length greater than 0.5 cMs exhibits very similar behavior as BEAGLE (0.5% decrease). Figure 2e shows the sensitivity and precision (i.e., positive predictive value – PPV) of the imputed genotypes with respect to changing impute genotype probability. For each window length, a curve is plotted. As the window length increases, both the sensitivity and precision increases. BEAGLE exhibits the optimal baseline

performance. Above 0.3 cMs, the curves are very close to each other. These results indicate that $l_w > 0.3$ cM is the minimum window length with comparable accuracy as BEAGLE. For uncommon variants, we observed that most of the R^2 and concordance are at the high or low accuracy regimes (Fig. 3a,b) for both BEAGLE and LoHaMMer. The PR curves demonstrate a fairly steady pattern of change in the accuracy (Fig. 3c,d).

Effective Population Size (N_e). We analyzed how effective population size impacts prediction accuracy. We tested effective population sizes using $N_e \in \{10^2, 10^3, 10^4, 10^6\}$. Figures 2f and 2g show the accuracy and sensitivity-precision curves, respectively. We observed that $N_e \approx 10^3$ provides the best performing population size. Also, increasing N_e too high renders the transition probabilities uniform among self-transitions and recombination events causing too many recombinations. In turn, the model tends to overfit into any haplotype data without too much discriminative power since the model allows a large number of recombinations without much penalty. This causes the low performance with increasing N_e . For uncommon variants, the impact of population size is very distinctive, $N_e \approx 10^3$ provides substantially higher imputation accuracy compared to other higher parameters (Fig 2h). The increase in N_e causes a sharp decline in imputation accuracy. For the uncommon variants, while a similar trend was observed, we also found out the ML method performs better for large values of N_e . This highlights that ML-based imputation can provide more robust imputation results for certain parameters (Fig. 3e-h).

Target-Center Distance (l_{c2t}). The positioning of the target variant, l_{c2t} (target-center distance), in the imputation window is another parameter that can impact imputation accuracy (Figure 1a). We tested the imputation of accuracy with increasing target-center distance values, $l_{c2t} \in \{0.02, 0.05, 0.1, 0.15\}$ cM. We use genetic distance as the measurement of unit for these parameters since it is the most natural choice. Imputation accuracy is shown for different center-target distance values indicating a visible impact of target-center distance (Fig 2i, 2j, 2k). We observed that the imputation accuracy decreases as l_{c2t} increases. This expected as smaller l_{c2t} implies that the haplotype evidence from the two sides of the target variant is optimized. Below $l_{c2t} < 0.1$ cM, we observed that the local window-based imputation provides comparable accuracy. For uncommon variants, we observed similar pattern in the accuracy (Figure 3i-l). As with other parameters, the rare variant are more sensitive to changes in target-center distance.

Maximum Number of Tag (Typed) Variants in Window (n_{tag}). The last set of parameters we tested are the number of tag (i.e., typed) variants that are used for imputation. For this, we subsampled the tags in each window such that the number of tag variants is bounded by the maximum number of tag variants. For this, we evaluated the impact of changing $n_{tag} \in \{10, 50, 100, 200, 1000\}$. The typed variants in the windows that harbor less than n_{tag} tag variants are used as they are. Figures 2k and 2m show the average R^2 and non-reference genotype concordance. For n_{tag} greater than 100 variants, we observed that the accuracy levels out with a slight increase, both for common. For rare variants, we observed that the accuracy flattens out around $n_{tag} = 200$. Similar conclusions can be seen from the sensitivity-PPV curve (Fig. 2n). For uncommon variants, we observed a similar trend (Fig. 3m-p).

Overall, these results demonstrate that the local-HMM computation can provide imputation accuracy comparable to the baseline methods, especially for the common variants.

Accuracy on Chromosomes 19 and 20

In order to validate and compare the local HMM parameter accuracy on a different positions, we tested the parameters for the variants on chromosomes 19 and 20. We extracted the typed variants on the Illumina Duo array platform on chromosomes 19 and 20. After this, we extracted 24,333 of 27,403 typed variants on chromosome 19 and 26,405 of the 28,319 typed variants on chromosome 20. The remaining variants (768,292 variants on chr19 and 742,370 on chr20) are used as untyped variants that are imputed by local HMM and by BEAGLE. We classified the variants with respect to MAF by separating variants into 4 different MAF ranges: 1) $MAF \in [0, 0.005]$ (Very rare), 2) $MAF \in [0.005, 0.01]$ (Rare), 3) $MAF \in [0.01, 0.05]$ (Uncommon), 4) $MAF \in [0.05, 0.5]$ (Common). Before imputing untyped variants, the genotypes are phased using Eagle2⁵¹. We use the parameters $(l_w, N_e, l_{c2t}, n_{tag}) = (0.5, 10^3, 0.02, 1000)$ for validation of accuracy. Figure 4 shows the R^2 and non-reference genotype concordance distribution for the variants on chromosome 19 (Fig. 4a,b) and chromosome 20 (Fig. 4c,d). The imputation of variants in the MAF range of common and uncommon are comparable with the baseline imputations of BEAGLE with less than 1% different in accuracy between baseline and local HMM. For rare and very rare variants, BEAGLE outperforms with more than 2 percent accurately in R^2 . The non-reference genotype concordance is less than 1 percent for these categories. These results indicate that local HMMs can potentially provide utility for uncommon and common variants (i.e., $MAF > 1\%$).

Discussion

We analyzed the feasibility of imputing variants using HMMs that are computed on locality of the target variants, i.e. untyped variants. There are several advantages of focusing exclusively to the locality of an untyped target. First, the computations can be parallelized and performed at a much smaller scale without the need of large number of untyped variants. Second, the evaluation of assessment of the local HMM accuracy can provide biological insight into the haplotype structure and imputability estimates⁵⁸. Third, the local models can be run in isolation from other parts of the genome. This way, the imputation algorithms can be re-designed for other tasks. For instance, recently developed privacy-aware imputation methods make extensive use of the locality-based models. Our results provide insight into the design of secure imputation algorithms. Also, our study provides evidence that HMM-based imputation methods can be designed with a locality-based approach.

It should be noted that the default parameters does not provide the optimal performance that can be achieved using local HMMs. For instance, we did not evaluate the impact of increasing l_w while the maximum number of tag variants is kept constant. This would still constitute a local HMM model since the maximum number of surrounding tags is constrained. In other words, this would keep the computational requirements constant but it would enable local HMM to assess large haplotype blocks. In addition, the locality windows can be implemented in different ways. For example, the tags can be filtered with respect to the smallest genetic distance, i.e., we can remove the tag variants that are close to each other.

The main limitation of the local-HMM methods that are evaluated here is the lower accuracy for rare variants, especially for the variants with MAF lower than 1%. Our results show that the performance can be improved by extending the local windows to include more variants. This is reasonable since longer windows enable the resolution of the rare haplotypes more accurately than shorter windows. From a utility perspective, we observed that most of the downstream analyses, such as genomewide association studies (GWAS) impose thresholds on the well above 1%. For instance, even high powered GWAS studies

impose thresholds at 2-5% on the MAF of the variants to provide enough power for detecting phenotype-genotype associations⁵⁹. Also, even the state-of-the-art HMM methods do not provide the imputation accuracy for low MAF variants that is necessary for the downstream analyses. In addition, these rare variants tend to be population-specific⁶⁰ and usage of population specific panels can enable more accurate performance. Thus, the local HMMs can be used to impute variants for downstream tasks with MAF values that are utilizable for studies such as GWAS.

Methods

We present the computational details of the ML and MAP estimation from the local-HMM.

Description of the Local-HMMs

LoHaMMer computes MAP and ML estimates on the typed tag variants, i.e., keeps track of haplotype paths that are passing through only the typed variants. We assume that the genotypes are phased and the genotype matrix is denoted by $G_{i,j}^{(h)}$, which stands for the allele on parental copy h for individual i and the variant at index j . The parental copy has two values $h \in \{0,1\}$, indicating the paternal and maternal haplotypes (or vice versa). $G_{i,[1,j]}^{(h)}$ indicates the sequence of alleles for i^{th} individual for variants between 1 and j , i.e., $G_{i,[1,j]}^{(h)} = (G_{i,1}^{(h)}, G_{i,2}^{(h)}, \dots, G_{i,j}^{(h)})$. The alleles for each variant can have 2 values, $G_{i,j}^{(h)} \in \{0,1\}$, denoting reference allele and alternate alleles. $S_{i,k}^{(h)}$ denotes the HMM state at the variant k for i^{th} individual. The states correspond to the indices of haplotypes in the phased reference genotype panel, i.e., $S_{i,k}^{(h)} \in [1, N]$. We denote the indices of the untyped variants with $j_\emptyset$, which is the set of variant indices (i.e., $j < V$) for which the genotypes are missing.

Variant Subsampling. Given the maximum number of typed variants (or tag variants), $\hat{n}_{tag}$, LoHaMMer first identifies all the variants in the current window, which is of length l_w . Given that $n_{tag} > \hat{n}_{tag}$ is the total number of variants, LoHaMMer takes every $\left(\frac{\hat{n}_{tag}}{n_{tag}}\right)^{th}$ variant to select $\hat{n}_{tag}$ in the window. If n_{tag} is smaller than $\hat{n}_{tag}$, all the tag variants are used for imputation. To simplify the presentation, we assume that the variant indexing is based on the subsampled variant list.

Posterior Probability Estimation by Forward-Backward Algorithm (MAP Method). The MAP algorithm relies on computation of forward and backward matrices. Given individual i and haplotype h , the forward probability is formulated as:

$$P\left(G_{i,[1,j]}^{(h)} \mid S_{i,j}^{(h)} = a\right) = \sum_{\substack{\forall S_{i,[1,j]} \\ S_{i,j}=a}} \left(\sum_{1 \leq k \leq j} \left(\underbrace{\tau_k(S_{i,k-1} \rightarrow S_{i,k})}_{\text{Transition from } S_{i,k-1} \text{ to } S_{i,k}} \times \underbrace{e_k(G_{i,k}^{(h)}, S_{i,k})}_{\text{Emission of } G_{i,k}^{(h)} \text{ from the state } S_{i,k}} \right) \right)$$

where $P\left(G_{i,[1,j]}^{(h)} \mid S_{i,j}^{(h)} = a\right)$ denotes the forward variable, which is the total probability of all state sequences and the emissions from the state sequences $S_{i,[1,j]}$ that emit the allele sequence $G_{i,[1,j]}^{(h)}$ with

the constraint that the last state at variant j is a , i.e., $S_{i,j}^{(h)} = a$. The forward variable matrix can be computed for all variant positions and all states using recursion:

$$P\left(G_{i,[1,j]}^{(h)} \mid S_{i,j}^{(h)} = a\right) = \sum_{1 \leq b \leq N} \left(\underbrace{P\left(G_{i,[1,j-1]}^{(h)} \mid S_{i,j-1}^{(h)} = b\right)}_{\text{Forward variable at } (j-1)} \times \tau_b(b \rightarrow a) \times e_j\left(G_{i,j}^{(h)}, a\right) \right)$$

where the forward variable at variant j is computed using the forward variable at position $(j - 1)$. The boundary condition is defined at the first nucleotide:

$$\forall a \in [1, N], \quad P\left(G_{i,[1,0]}^{(h)} \mid S_{i,0}^{(h)} = a\right) = \frac{1}{N}$$

which indicates that the state at the first variant is uniformly distributed among all states, i.e. there is no preference between haplotypes that initiate the HMM. This boundary condition is sometimes described by introducing a special state named the “start state”.

The backward probability is formulated as:

$$P\left(G_{i,[j,V]}^{(h)} \mid S_{i,j} = a\right) = \sum_{\substack{\forall S_{i,[j,V]} \\ S_{i,j}=a}} \left(\sum_{j \leq k \leq V} \left(\underbrace{\tau_k(S_{i,k} \rightarrow S_{i,k+1})}_{\text{Transition from } S_{i,k} \text{ to } S_{i,k+1}} \times \underbrace{e_k\left(G_{i,k}^{(h)}, S_{i,k}\right)}_{\substack{\text{Emission of } G_{i,k}^{(h)} \\ \text{from the state } S_{i,k}}} \right) \right)$$

denotes the backward matrix for i^{th} individual's haplotype h , and the total probability over all the state sequences, $S_{i,[j,V]}$, that emit the allele subsequence $G_{i,[j,V]}^{(h)}$ with the constraint that the first state at variant j is a , i.e., $S_{i,j}^{(h)} = a$. The backward variable can be computed using a simple recursion relationship, similar to the forward variable:

$$P\left(G_{i,[j,V]}^{(h)} \mid S_{i,j}^{(h)} = a\right) = \sum_{1 \leq b \leq N} \left(\underbrace{P\left(G_{i,[j+1,V]}^{(h)} \mid S_{i,j+1}^{(h)} = b\right)}_{\text{Backward variable at } (j+1)} \times \tau_j(a \rightarrow b) \times e_j\left(G_{i,j}^{(h)}, a\right) \right)$$

with the boundary conditions:

$$\forall a \in [1, N], \quad P\left(G_{i,[V+1,V]}^{(h)} \mid S_{i,V+1}^{(h)} = a\right) = \frac{1}{N}$$

which indicates that the haplotypes are uniformly distributed at the end of the allele sequence.

Computation of the Posterior Probability for Untyped Variants. The forward and backward variables are used for inferring the probability of the observing alleles 0 and 1 at the untyped variants. To estimate the allele probabilities of an untyped variant at index $j \in J_\emptyset$, LoHaMMer identifies the two consecutive typed

variants that are closest to the variant j . Using the nearest tags, LoHaMMer uses an approach similar to BEAGLE to estimate the path that passes along the untyped variant:

$$\forall j \in j_{\emptyset}, P(G_{i,j}^{(h)} = t) = \sum_{\substack{1 \leq a \leq N \\ H_{a,j} = t}} \left(P(G_{i,[1,k]}^{(h)} | S_{i,k}^{(h)} = a) \times P(G_{i,[k+1,V]}^{(h)} | S_{i,k+1}^{(h)} = a) \right), t \in \{0,1\}$$

where $j_{\emptyset}$ indicates the untyped variant indices in the genotype matrix, and k is the variant index such that variants at k and $k + 1$ are the closest typed tags to untyped variant j . The allelic probabilities from the parental copies are normalized and combined to generate a final genotype probability for the 3 possible genotypes:

$$\begin{aligned} P(G_{i,j} = 0) &= P'(G_{i,j}^{(0)} = 0) \times P'(G_{i,j}^{(1)} = 0) \\ P(G_{i,j} = 1) &= P'(G_{i,j}^{(0)} = 0) \times P'(G_{i,j}^{(1)} = 1) + P'(G_{i,j}^{(0)} = 1) \times P'(G_{i,j}^{(1)} = 0) \\ P(G_{i,j} = 2) &= P'(G_{i,j}^{(0)} = 1) \times P'(G_{i,j}^{(1)} = 1) \end{aligned}$$

where $P'(G_{i,j}^{(0)} = 1)$ denotes the normalized probabilities that are in the range $[0,1]$. These are computed as following

$$P'(G_{i,j}^{(h)} = t) = \frac{P'(G_{i,j}^{(h)} = t)}{P'(G_{i,j}^{(h)} = t) + P'(G_{i,j}^{(h)} = 1 - t)}$$

where the normalization is performed over the two possible allelic probabilities for the parental copy t .

Maximum-Likelihood Haplotype Path Estimation by Viterbi Algorithm (ML Method). Similar to the forward matrices, the ML method keeps track of the maximum scoring matrix at each typed tag for every possible haplotype state:

$$P^{\wedge}(G_{i,[1,j]}^{(h)} | \check{S}_{i,j}^{(h)} = a) = \max_{\substack{S_{i,[1,j]}^{(h)} \\ S_{i,j}^{(h)} = a}} \left(\prod_{1 \leq k \leq j} \left(\underbrace{\tau_k(S_{i,k-1} \rightarrow S_{i,k})}_{\text{Transition from } S_{i,k-1} \text{ to } S_{i,k}} \times \underbrace{e_k(G_{i,k}^{(h)}, S_{i,k})}_{\text{Emission of } G_{i,k}^{(h)} \text{ from the state } S_{i,k}} \right) \right)$$

where $P^{\wedge}(G_{i,[1,j]}^{(h)} | \check{S}_{i,j}^{(h)} = a)$ indicates the probability of the typed allele sequence $G_{i,[1,j]}^{(h)}$ from the most likely state sequence $\check{S}_{i,[1,j]}^{(h)}$ with additional constraint of $\check{S}_{i,j}^{(h)} = a$. This path is the ML path that LoHaMMer uses to infer the most likely haplotype mosaic that emits the typed allele sequence. This array is computed using the following recursion relationship:

$$P^{\wedge} \left(G_{i,[1,j]}^{(h)} \mid \check{S}_{i,j}^{(h)} = a \right) = \max_{1 \leq b \leq N} \left(\underbrace{P \left(G_{i,[1,j-1]}^{(h)} \mid \check{S}_{i,j-1}^{(h)} = b \right)}_{\text{ML variable at } (j-1)} \times \tau_b(b \rightarrow a) \times e_j \left(G_{i,j}^{(h)}, a \right) \right)$$

LoHaMMer computes the ML matrix using this recursion relationship for every typed tag variant from left to right for all the haplotypes with the boundary condition:

$$\forall a \in [1, N], \quad P^{\wedge} \left(G_{i,[1,0]}^{(h)} \mid \check{S}_{i,0}^{(h)} = a \right) = \frac{1}{N}$$

As for the forward and backward matrices, the ML matrix is computed over the typed tag variants.

After computing the ML matrix, LoHaMMer traces back the ML matrix to identify the optimal state sequence i.e., the optimal set of haplotypes that emits the full allelic sequence:

$$\check{S}_{i,[1,V]}^{(h)} = \operatorname{argmax}_{S_{[1,V]}} \left\{ P \left(G_{i,[1,V]}^{(h)} \mid S_{[1,V]} \right) \right\}.$$

After the optimal state is assigned, LoHaMMer assigns the alleles to the untyped variants similar to the MAP method. For the untyped variant at index j , LoHaMMer identifies the closest typed tag variant and assigns the allele based on the state on the tag variant:

$$\forall j \in j_{\emptyset}, \quad \check{G}_{i,j}^{(h)} = H_{t,k} ; t = \check{S}_{i,j}^{(h)}$$

where k is the tag variant that is closest to the untyped variant at j .

Haplotype Clustering in Blocks of Variants. The recursion relationships for ML and forward-backward variants indicate that it is necessary to perform a summation (or a maximum operation) over all the haplotypes in the reference panel, for every typed tag variant. This computation can become quickly intractable. Similar to the previous methods, LoHaMMer clusters the haplotypes, computes each MAP and ML arrays over the clusters of reference haplotypes to minimize the number of redundant operations. The clustering increases the efficiency substantially because (1) the number of unique haplotypes over short stretches increase much slower compared to the number of haplotypes, (2) the transition probabilities between states depends only on the self-transition and recombinations. These optimizations are extensively described in previous methods.

We will briefly describe the usage of clustering for computation of ML arrays. LoHaMMer selects a number of variants that will be used to cluster the reference haplotypes, by default the block length is selected to be 10 variants. Given a local window, LoHaMMer divides the window into blocks of 10 variants. Next the reference haplotypes on each block are clustered such that each cluster corresponds to a unique sequence of 10 alleles, corresponding to 10 variants in the block. Next, for each cluster, the ML variable is computed as the maximum of the ML variable over the haplotypes in the cluster. Since the clusters share the allelic sequence exactly, ML variables for the clusters are computed at the cluster-level using the recursion relationships over the 10 variants in the block. After cluster-level ML variables are computed for each cluster, LoHaMMer assigns the ML variable to each haplotype from their corresponding cluster-level ML variables.

Numerical Stability. The transition and emission probabilities are smaller than 1 and they are multiplied with each other over all transitions and emissions. Thus, the ML variable and MAP variables may overrun or underrun the numerical precision. To get around these numerical stability issues, LoHaMMer can perform the computations in the logarithmic domain or it scales the ML and forward-backward variables by a scaling factor. For the logarithmic domain computations, LoHaMMer keeps every value as logarithms. In logarithmic domain, a multiplication is converted to a summation and this is convenient since the overflow is virtually impossible. However, we observed that the approximate summation in logarithmic domain requires numerous slow operations and causes big decreases in speed. Therefore, LoHaMMer uses a linear scaling value by default. For this, LoHaMMer multiplies every array value by a constant scaling factor. We observed by trial-and-error that scaling factor of $\exp(0.2)$ enables minimal number of underflow or overflow issues. LoHaMMer keeps track of any overflow and underflow at each computation step. If an array value becomes too high or too low, the values are re-scaled to ensure numerical stability.

Computation of Accuracy Metrics

Genotype R^2 . The genotype R^2 is computed as the squared value of the Pearson correlation coefficient between the known and imputed genotype for each variants. For the j^{th} untyped variant the genotype R^2 is computed as:

$$R_j^2 = \left(\sigma(G_{.,j}, G_j^{Known}) \right)^2$$

where σ indicates the Pearson correlation of the genotype value.

Non-Reference Genotype Concordance. The concordance is simply the overlap between the genotypes that are known to be non-reference:

$$\kappa_j^{(NR)} = \frac{|\{i \mid G_{i,j}^{Known} > 0\} \cap G_{.,j}|}{|\{i \mid G_{i,j}^{Known} > 0\}|}$$

where $\kappa_j^{(NR)}$ denotes the non-reference concordance between the known non-reference genotypes of variant j and the imputed variants over all individuals.

Data Sources

The 1000 genomes project genotypes are downloaded from NCBI ftp data portal. The Illumina Duo v3 variants are extracted from the array's documentation available at: ftp://webdata2.webdata2@ussd-ftp.illumina.com/MyIllumina/c1859532-43c5-4df2-a1b4-c9f0374476d2/Human1M-Duov3_H.csv. The variants in The 1000 Genomes Project that overlap with the variants on the array's tag variants are used as the typed variants.

Figure Legends

Figure 1. (a) Illustration of the local imputation setup. Query individual's parental haplotype copies are shown in two rectangles in top, which are strings of {0,1}. 0 and 1 indicate a reference and alternate allele, respectively, for corresponding variants. The genetic distance (in centimorgans) are shown with the blue arrow and is used to track the center position of the window and the target untyped variant. The reference haplotypes are shown in the box below wherein each row corresponds to a haplotype. Given the local window of radius l_{win} the window is illustrated in the dashed rectangle whose center is shown at the

genetic position c and for the target variant at position t . The typed variants are shown in green rectangles and the untyped target is shown in the red rectangle, whose alleles on the query haplotypes are shown with question marks. (b) Illustration of the forward and backward variables. For the 3rd haplotypes at the untyped variant, the incoming paths (forward variable) are illustrated with green lines. The outgoing transitions are shown with red lines. The self-transitions are shown with heavier lines compared to the non-self transitions. (c) The ML (Viterbi) path is shown with the transitions along the haplotypes.

Figure 2. (a) The distribution of typed variant number for different window lengths for common untyped variants. (b) The number of tag variants for 1 cM over the chromosomes, each row is a chromosome and x-axis shows the chromosomal position. (c,d) Distribution of genotype R^2 and genotype concordance for changing locality window length (l_{win}). (e) The PR-curve for changing l_{win} . (f,g) Distribution of genotype R^2 and genotype concordance for changing population size (N_e) (h) PR-curve for changing N_e . (i,j) Distribution of genotype R^2 and genotype concordance for changing target-center distance, l_{c2t} . (k) PR-curve for changing l_{c2t} . (l,m) Distribution of genotype R^2 and genotype concordance for different number of targets. (n) PR-curve for different number of targets.

Figure 3. (a,b) Distribution of genotype R^2 and genotype concordance for changing locality window length (l_{win}). (c,d) The PR-curve (all and non-reference) for changing l_{win} . (e, f) Distribution of genotype R^2 and genotype concordance for changing population size (N_e) (g, h) PR-curves for changing N_e . (i, j) Distribution of genotype R^2 and genotype concordance for changing target-center distance, l_{c2t} . (k, l) PR-curves for changing l_{c2t} . (m, n) Distribution of genotype R^2 and genotype concordance for different number of targets. (o, p) PR-curve for different number of targets.

Figure 4. (a) Distribution of genotype R^2 and **(b)** non-reference genotype concordance for the untyped variants on chromosome 19. The variants are stratified with respect to minor allele frequency (MAF) as shown on the x-axis. **(a)** Distribution of genotype R^2 and **(b)** non-reference genotype concordance for the untyped variants on chromosome 20.

References

1. Goldfeder, R. L., Wall, D. P., Khoury, M. J., Ioannidis, J. P. A. & Ashley, E. A. Human Genome Sequencing at the Population Scale: A Primer on High-Throughput DNA Sequencing and Analysis. *Am. J. Epidemiol.* **186**, 1000–1001 (2017).
2. Rehm, H. L. Evolving health care through personal genomics. *Nature Reviews Genetics* **18**, 259–267 (2017).
3. Burn, J. Should we sequence everyone's genome? Yes. *Bmj* **3133**, 2–3 (2013).
4. Schloss, J. A., Gibbs, R. A., Makhijani, V. B. & Marziani, A. Cultivating DNA Sequencing Technology After the Human Genome Project. *Annu. Rev. Genomics Hum. Genet.* **21**, 117–138 (2020).
5. Pepke, S., Wold, B. & Mortazavi, A. Computation for ChIP-seq and RNA-seq studies. *Nat. Methods* **6**, S22–S32 (2009).
6. Bycroft, C. *et al.* Genome-wide genetic data on ~500,000 UK Biobank participants. *bioRxiv* (2017). doi:10.1101/166298
7. Collins, F. S. & Varmus, H. A new initiative on precision medicine. *N. Engl. J. Med.* (2015). doi:10.1056/NEJMp1500523

8. Chisholm, J., Caulfield, M., Parker, M., Davies, J. & Palin, M. Briefing- Genomics England and the 100K Genome Project. *Genomics England* (2013). Available at: <http://www.genomicsengland.co.uk/briefing/>.
9. Li, M. M. *et al.* Standards and Guidelines for the Interpretation and Reporting of Sequence Variants in Cancer: A Joint Consensus Recommendation of the Association for Molecular Pathology, American Society of Clinical Oncology, and College of American Pathologists. *J. Mol. Diagn.* **19**, 4–23 (2017).
10. Green, R. C. *et al.* ACMG recommendations for reporting of incidental findings in clinical exome and genome sequencing. *Genet. Med.* (2013). doi:10.1038/gim.2013.73
11. Guerrini, C. J. & Contreras, J. L. Credit for and Control of Research Outputs in Genomic Citizen Science. *Annual Review of Genomics and Human Genetics* **21**, 465–489 (2020).
12. Evans, J. P. Recreational genomics; what's in it for you? *Genet. Med.* **10**, 709–710 (2008).
13. Bergström, A. *et al.* Insights into human genetic variation and population history from 929 diverse genomes. *Science* (80-.). **367**, (2020).
14. Sudlow, C. *et al.* UK Biobank: An Open Access Resource for Identifying the Causes of a Wide Range of Complex Diseases of Middle and Old Age. *PLoS Med.* **12**, (2015).
15. Moscarello, T., Murray, B., Reuter, C. M. & Demo, E. Direct-to-consumer raw genetic data and third-party interpretation services: more burden than bargain? *Genet. Med.* **21**, 539–541 (2019).
16. Guglielmi, G. Facing up to injustice in genome science. *Nature* **568**, 290–293 (2019).
17. Aguilar-Ordoñez, I. *et al.* Whole genome variation in 27 Mexican indigenous populations, demographic and biomedical insights. *PLoS One* **16**, e0249773 (2021).
18. Caron, N. R. *et al.* Indigenous Genomic Databases: Pragmatic Considerations and Cultural Contexts. *Front. Public Heal.* **8**, 111 (2020).
19. Genome As Commodity - IEEE Spectrum. Available at: <https://spectrum.ieee.org/biomedical/diagnostics/genome-as-commodity>. (Accessed: 27th September 2019)
20. Telenti, A. & Jiang, X. Treating medical data as a durable asset. *Nature Genetics* **52**, 1005–1010 (2020).
21. NHGRI. Genomic Data Sharing Policy. *National Human Genome Research Institute* (2019). Available at: <https://www.genome.gov/about-nhgri/Policies-Guidance/Genomic-Data-Sharing>. (Accessed: 10th June 2021)
22. The Lancet. Genomic sequencing in pandemics. *The Lancet* **397**, 445 (2021).
23. Torkamani, A., Wineinger, N. E. & Topol, E. J. The personal and clinical utility of polygenic risk scores. *Nature Reviews Genetics* (2018). doi:10.1038/s41576-018-0018-x
24. Bush, W. S. & Moore, J. H. Chapter 11: Genome-Wide Association Studies. *PLoS Comput. Biol.* **8**, (2012).
25. Schaid, D. J., Chen, W. & Larson, N. B. From genome-wide associations to candidate causal variants by statistical fine-mapping. *Nature Reviews Genetics* **19**, 491–504 (2018).

26. Sella, G. & Barton, N. H. Thinking About the Evolution of Complex Traits in the Era of Genome-Wide Association Studies. *Annu. Rev. Genomics Hum. Genet.* **20**, 461–493 (2019).
27. Stram, D. O. Tag SNP selection for association studies. *Genetic Epidemiology* **27**, 365–374 (2004).
28. Fallin, D. & Schork, N. J. Accuracy of haplotype frequency estimation for biallelic loci, via the expectation-maximization algorithm for unphased diploid genotype data. *Am. J. Hum. Genet.* **67**, 947–959 (2000).
29. Sboner, A., Mu, X., Greenbaum, D., Auerbach, R. K. & Gerstein, M. B. The real cost of sequencing: higher than you think! *Genome Biology* **12**, 125 (2011).
30. Illumina Announces a New Genotyping Array and Scientific Contribution to Three Genome Centers to Support the All of Us Research Program. Available at: <https://www.illumina.com/company/news-center/press-releases/2018/2379861.html>. (Accessed: 5th October 2020)
31. Das, S., Abecasis, G. R. & Browning, B. L. Genotype Imputation from Large Reference Panels. *Annu. Rev. Genomics Hum. Genet.* **19**, 73–96 (2018).
32. Yun, L., Willer, C., Sanna, S. & Abecasis, G. Genotype imputation. *Annual Review of Genomics and Human Genetics* **10**, 387–406 (2009).
33. The International HapMap 3 Consortium. Integrating common and rare genetic variation in diverse human populations. *Nature* **467**, 52–8 (2010).
34. Thompson, E. A. Identity by descent: Variation in meiosis, across genomes, and in populations. *Genetics* **194**, 301–326 (2013).
35. Browning, S. R. & Browning, B. L. Identity by descent between distant relatives: Detection and applications. *Annual Review of Genetics* **46**, (2012).
36. Zhou, Y., Browning, S. R. & Browning, B. L. A Fast and Simple Method for Detecting Identity-by-Descent Segments in Large-Scale Data. *Am. J. Hum. Genet.* **106**, 426–437 (2020).
37. Naseri, A., Liu, X., Tang, K., Zhang, S. & Zhi, D. RaPID: ultra-fast, powerful, and accurate detection of segments identical by descent (IBD) in biobank-scale cohorts. *Genome Biol.* **20**, 143 (2019).
38. Rabiner, L. R. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proc. IEEE* **77**, 257–286 (1989).
39. Yoon, B.-J. Hidden Markov Models and their Applications in Biological Sequence Analysis. *Curr. Genomics* **10**, 402–415 (2009).
40. Li, N. & Stephens, M. Modeling Linkage Disequilibrium and Identifying Recombination Hotspots Using Single-Nucleotide Polymorphism Data. *Genetics* **165**, 2213–2233 (2003).
41. Browning, B. L., Zhou, Y. & Browning, S. R. A One-Penny Imputed Genome from Next-Generation Reference Panels. *Am. J. Hum. Genet.* **103**, 338–348 (2018).
42. Rubinacci, S., Delaneau, O. & Marchini, J. Genotype imputation using the Positional Burrows Wheeler Transform. *PLoS Genet.* **16**, (2020).
43. Das, S. *et al.* Next-generation genotype imputation service and methods. *Nat. Genet.* **48**, 1284–1287 (2016).

44. 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).
45. Van Leeuwen, E. M. *et al.* Population-specific genotype imputations using minimac or IMPUTE2. *Nat. Protoc.* **10**, 1285–1296 (2015).
46. Taliun, D. *et al.* Sequencing of 53,831 diverse genomes from the NHLBI TOPMed Program. *Nature* **590**, 290–299 (2021).
47. Huang, J. *et al.* Improved imputation of low-frequency and rare variants using the UK10K haplotype reference panel. *Nat. Commun.* (2015). doi:10.1038/ncomms9111
48. Chen, J. & Shi, X. Sparse convolutional denoising autoencoders for genotype imputation. *Genes (Basel)*. **10**, (2019).
49. Kojima, K. *et al.* A genotype imputation method for de-identified haplotype reference information by using recurrent neural network. *PLoS Comput. Biol.* **16**, (2020).
50. Kim, M. *et al.* Ultra-Fast Homomorphic Encryption Models enable Secure Outsourcing of Genotype Imputation. *bioRxiv* (2020).
51. Loh, P. R. *et al.* Reference-based phasing using the Haplotype Reference Consortium panel. *Nat. Genet.* **48**, 1443–1448 (2016).
52. Bahl, L. R., Cocke, J., Jelinek, F. & Raviv, J. Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate. *IEEE Trans. Inf. Theory* **20**, 284–287 (1974).
53. Forney, G. D. The Viterbi Algorithm. *Proc. IEEE* **61**, 268–278 (1973).
54. Skelly, D. A., Magwene, P. M. & Stone, E. A. Sporadic, global linkage disequilibrium between unlinked segregating sites. *Genetics* **202**, 427–437 (2016).
55. The 1000 Genomes Project Consortium. A global reference for human genetic variation. *Nature* **526**, 68–74 (2015).
56. Frazer, K. A. *et al.* A second generation human haplotype map of over 3.1 million SNPs. *Nature* **449**, 851–861 (2007).
57. Browning, B. L. & Browning, S. R. A Unified Approach to Genotype Imputation and Haplotype-Phase Inference for Large Data Sets of Trios and Unrelated Individuals. doi:10.1016/j.ajhg.2009.01.005
58. Duan, Q., Liu, E. Y., Croteau-Chonka, D. C., Mohlke, K. L. & Li, Y. A comprehensive SNP and indel imputability database. *Bioinformatics* **29**, 528–531 (2013).
59. Sung, Y. J. *et al.* A Large-Scale Multi-ancestry Genome-wide Study Accounting for Smoking Behavior Identifies Multiple Significant Loci for Blood Pressure. *Am. J. Hum. Genet.* (2018). doi:10.1016/j.ajhg.2018.01.015
60. Mitt, M. *et al.* Improved imputation accuracy of rare and low-frequency variants using population-specific high-coverage WGS-based imputation reference panel. *Eur. J. Hum. Genet.* **25**, 869–876 (2017).

LoHaMMer: Local Hidden Markov Models for Phased Genotype Imputation

Figure 1a: Local HMM Parameters

Figure 1b: MAP probability

Figure 1c: ML Solutions

Figure 2: Tag variant statistics

Figure 2: win_cM: R2, Geno Conc, PR stats – chr22 / common

Figure 2e: win_cM: R2, PR stats – chr22 / common

Figure 2: N_e: R2, PR stats – chr22 / common

Figure 2: N_e: R2, PR stats – chr22 / common

Figure 2: maximum Target-Center Distance: R2 – chr22 / common

Figure 2: maximum target-center distance: PR stats – chr22 / common

Figure 2: Max. # Tags: R2 – chr22 / common

Figure 2: Max # Tags: PR stats – chr22 / common

Figure 3: win_cM: R2 – chr22 / rare

Figure 3: win_cM: PR – chr22 / rare

Figure 3: N_e: R2 – chr22 / rare

Figure 3: N_e: PR – chr22 / rare

Figure 3: Center-Target Distance: R2 – chr22 / rare

Figure 3: Center-Target Distance : PR – chr22 / rare

Figure 3: Max # Tags: R2 – chr22 / Rare

Figure 3: Max # Tags: PR – chr22 / rare

Figure 4: Chr19 : R2 / NR-Conc. stats

Figure 4: Chr20 : R2 / NR-Conc. stats