Jin and Terhorst
*Algorithms for Molecular Biology* (2023) 18:12
https://doi.org/10.1186/s13015-023-00237-z

Algorithms for Molecular Biology

RESEARCH

Open Access

# The solution surface of the Li-Stephens haplotype copying model

Yifan Jin[1] and Jonathan Terhorst[1*]

**Abstract**

The Li-Stephens (LS) haplotype copying model forms the basis of a number of important statistical inference procedures in genetics. LS is a probabilistic generative model which supposes that a sampled chromosome is an imperfect mosaic of other chromosomes found in a population. In the frequentist setting which is the focus of this paper, the output of LS is a "copying path" through chromosome space. The behavior of LS depends crucially on two user-specified parameters, $\theta$ and $\rho$, which are respectively interpreted as the rates of mutation and recombination. However, because LS is not based on a realistic model of ancestry, the precise connection between these parameters and the biological phenomena they represent is unclear. Here, we offer an alternative perspective, which considers $\theta$ and $\rho$ as tuning parameters, and seeks to understand their impact on the LS output. We derive an algorithm which, for a given dataset, efficiently partitions the $(\theta, \rho)$ plane into regions where the output of the algorithm is constant, thereby enumerating all possible solutions to the LS model in one go. We extend this approach to the "diploid LS" model commonly used for phasing. We demonstrate the usefulness of our method by studying the effects of changing $\theta$ and $\rho$ when using LS for common bioinformatic tasks. Our findings indicate that using the conventional (i.e., population-scaled) values for $\theta$ and $\rho$ produces near optimal results for imputation, but may systematically inflate switch error in the case of phasing diploid genotypes.

**Keywords** Li-Stephens model, Haplotype copying model, Solution path

## Background

Statistical analysis in genetics often requires evaluating the likelihood of a sample of genomes under a model of evolution. Unfortunately, this computation can rarely be performed exactly, because it requires integrating over the astronomical number of possible ancestry scenarios that could have generated the data. In 2003, Na Li and Matthew Stephens [1] proposed to approximate this intractable likelihood by modeling a newly sampled chromosome as a perturbed mosaic those previously observed. Simple yet effective, the Li-Stephens (LS) haplotype copying model has had a lasting impact in genetics and bioinformatics, with important applications to genotype imputation, phasing, linkage mapping, detecting nection, and other areas [2].

LS depends on two parameters, $\theta$ and $\rho$, which are usually interpreted as the rates of mutation and recombination per unit time. Curiously, however, the model is not cognizant of time: in genealogical terms, it assumes that the sampled chromosome finds common ancestry with any other member of the population at a pre-determined number of generations in the past [3]. Since, in real data, there will be wide variation in the of age of ancestry at different locations in the genome, the interpretation of $\theta$ and $\rho$, and their effect on inference, is not altogether clear—a fact which Li and Stephens acknowledged in their original paper.

*Correspondence:
Jonathan Terhorst
jonth@umich.edu
[1] Department of Statistics, University of Michigan, 1085 South University Avenue, Ann Arbor, MI 48103, USA

In this study, we explore an alternative, non-biological perspective of $\theta$ and $\rho$, choosing to view them instead as tuning parameters in a machine learning algorithm. Then, the salient objective becomes understanding their effect on the output of the LS model. We derive a new, efficient algorithm for determining the complete solution surface of both the haploid and diploid variants of the LS algorithm. That is, for a given data set, the algorithm partitions the $(\theta, \rho)$ plane into regions such that the output of LS is constant within each region.

Our algorithm can be viewed as characterizing the trade-off between the effects of recombination and mutation: as the ratio $\rho/\theta$ tends to zero, recombinations become increasingly less likely, and the LS model simply copies from the most closely related haplotype in its entirety, at the potential expense of many mismatches. Conversely, as $\rho/\theta \to \infty$, there is free recombination between neighboring markers, and the LS model is able to find a path which is identical-by-state at every position (assuming no alleles are private to the focal haplotype), at the expense of improbably many recombinations. Our contribution is to characterize the behavior of LS for all intermediate values of $\rho/\theta$ as well, using an efficient procedure that requires only a single pass over the data.

For readers who are familiar with $\ell_1$-regularized regression (the LASSO), this can be seen as a type of LARS [4] or solution-path algorithm for the LS model. Solution-path algorithms for the LASSO are widely used in bioinformatics, for example to analyze expression data [5], estimate survival curves [6], detect DNA copy number alterations [7], or infer gene regulatory networks [8]. Of course, the LASSO is regression, whereas the haplotype estimation problem addressed by LS is strictly unsupervised in practice. However, by applying our method to simulated data where the ground-truth ancestry is known, we can gain better insight into how the LS model functions, which can then be transferred to real-world applications.

#### Notation and definitions

We now define the LS model and introduce our algorithms. We note once and for all that here we focus squarely on the *frequentist* variant of LS, which returns a copying path (or pair of them, in the diploid algorithm) through haplotype space. The copying path(s) are obtained by running the Viterbi decoding algorithm to obtain the maximum *a posteriori* (MAP) hidden state path through a hidden Markov model. Some other formulations of the LS model adopt a Bayesian perspective, where uncertainty in the unobserved copying path is modeled via a posterior distribution over hidden copying states. The techniques we introduce here are not applicable in the Bayesian setting, since they characterize the way in which the MAP path of the LS model changes as $\theta$ and $\rho$ vary.

LS is used to decode positional ancestry of a "focal" chromosome consisting of $L$ linked markers, using a panel of $N$ "template" chromosomes. Each chromosome may be represented as a *haplotype*, that is a vector in $\mathcal{D}^L$, where $\mathcal{D} = \{\mathtt{a}, \mathtt{c}, \mathtt{g}, \mathtt{t}\}$ represent the four DNA nucleotides. The template haplotypes can be organized into a matrix

$$\mathbf{H} = (H_{\ell,n})_{\ell=1,\ldots,L}^{n=1,\ldots,N} \in \mathcal{D}^{L\times N}.$$

Throughout the paper, the variable $h \in \mathcal{D}^L$ will be used to refer to a generic focal haplotype, and similarly the letter $g \in \mathcal{D}^{L\times 2}$ is used to denote a generic *diplotype*, that is a sequence of (unphased) diploid genotypes. We consider $h$, $g$ and $\mathbf{H}$ as fixed instances of the above quantities, and will omit notational dependence on them when there is no chance of confusion.

For a positive integer $z$, the set $\{1, 2, \ldots, z\}$ is denoted by $[z]$. A *path (of length $\ell$)* is a sequence $\pi = (\pi_1, \ldots, \pi_\ell) \in [N]^\ell$ which characterizes the haplotype in $\mathbf{H}$ from which $h$ copies at each position $1, \ldots, \ell$. The notation $|\pi|$ is used to denote the length of a path, so $|\pi| = \ell$ for a path of length $\ell$.

Given a path $\pi$, the function

$$k(\pi) \stackrel{\text{def}}{=} \sum_{k=2}^{|\pi|} \mathbf{1}\{\pi_k \neq \pi_{k-1}\} \tag{1}$$

counts the number of times that $\pi$ switches templates (i.e., the number of crossover recombinations). Similarly, the function

$$m(\pi) := \sum_{k=1}^{|\pi|} \mathbf{1}_{\{h_k \neq H_{k,\pi_k}\}} \tag{2}$$

counts the number of mismatches between haplotype $h$ and $\mathbf{H}$ for the copying path $\pi$. In the diploid case, if $\pi$ and $\lambda$ are two copying paths of equal length, then

$$m(\pi, \lambda) = \sum_{k=1}^{|\pi|} \left|\{g_{k,1}, g_{k,2}\} \triangle \{H_{k,\pi_k}, H_{k,\lambda_k}\}\right|, \tag{3}$$

where $A \triangle B$ denotes the symmetric difference between sets $A$ and $B$, is the number of panel mismatches for the focal diplotype $g$. (Note that $m(\pi)$ and $m(\pi, \lambda)$ have implicit dependencies on $h$ and $g$ which have been suppressed for clarity.)

In the next sections, we will use some shorthand notation to refer to qualified subsets of the space of copying paths. A copying path $\pi$ is an *$\ell$-path* if $|\pi| = \ell$. An $\ell$-path for which $k(\pi) = r$ is an *$(\ell, r)$-path*, and similarly an

$(\ell, n)$-path is an $\ell$ path with the additional property that $\pi_\ell = n$. Lastly, an $(\ell, r, n)$-*path* meets all three of these criteria.

### The LS model

In its original formulation, LS is a generative model of the haplotype $h$ conditional on the template set $\mathbf{H}$. Formally, it is a hidden Markov model: at each position, $h$ selects a particular template $\pi_\ell \in [N]$ from $\mathbf{H}$, whose identity is latent and unobservable. Conditional on this selection, the template allele $H_{\ell,\pi_\ell}$ is faithfully copied to $h$, except with some small error probability $p_\theta$. The "copying path" $\pi \in [N]^L$ follows a stationary Markov chain: conditional on $\pi_{\ell-1}$, a switch occurs with probability $p_\rho \ll 1/N$; otherwise, with probability

$$1 - Np_\rho \tag{4}$$

there is no switch and $\pi_\ell = \pi_{\ell-1}$. The leading factor $N$ in (4) reflects the fact that, conditional on a switch having occurred between positions $\ell - 1$ and $\ell$, the identity of the newly selected haplotype at position $\ell$ is uniformly distributed among the $N$ possible panel haplotypes. Similarly, the probability of correctly copying is $1 - 3p_\theta$, where, again, the factor of 3 implies that the position mutates uniformly at random to one of the three other nucleotides not possessed by the template haplotype whenever a copying error occurs.

Thus, for a given $\pi$, the conditional likelihood of $h$ is

$$p(h \mid \pi, \mathbf{H}, \theta, \rho) \propto p_\rho^{k(\pi)}(1 - Np_\rho)^{L-k(\pi)-1} p_\theta^{m(\pi)}(1 - 3p_\theta)^{L-m(\pi)},$$

which leads to a compact expression for the negative log-likelihood [9]:

$$-\log p(h \mid \pi, \mathbf{H}, \theta, \rho) = \alpha(\theta)m(\pi) + \beta(\rho)k(\pi) + C, \tag{5}$$

where $C$ is a constant which does not depend on $\pi$, and we defined

$$\begin{aligned} \alpha(\theta) &\stackrel{\text{def}}{=} -\log \frac{p_\theta}{1 - 3p_\theta} \\ \beta(\rho) &\stackrel{\text{def}}{=} -\log \frac{p_\rho}{1 - Np_\rho}. \end{aligned} \tag{6}$$

The function $\mathrm{LS}_h(\theta, \rho)$ is defined to return the lowest possible cost for (5):

$$\mathrm{LS}_h(\theta, \rho) \stackrel{\text{def}}{=} \max_\pi \log p(h \mid \pi, \mathbf{H}, \theta, \rho). \tag{7}$$

Li and Stephen's original model is recovered by setting

$$p_\rho = \frac{1 - \exp(-\rho/N)}{N} \tag{8}$$

and $p_\theta = \tilde{\theta}/[2(N + \tilde{\theta})]$, where the constant $\tilde{\theta}$ is derived by a population genetic argument [1, eq. A3]. An alternative parameterization, based on a later, genealogical interpretation of LS [3], is to set

$$p_\theta = \frac{1 - \exp(-\theta/N)}{3}, \tag{9}$$

since the time to first coalescence between the focal and template haplotypes is roughly $1/N$ for large $N$. In general, different choices of $p_\rho$ and $p_\theta$ are possible, which may not have any genetic or biological interpretation. The perspective we adopt here is to treat them as numerical parameters, and try to understand their effect on the output of the LS algorithm. To that end, while it is technically possible for $\alpha(\theta)$ or $\beta(\rho)$ to be negative in (5), this requires very high rates of mutation and/or recombination which are not encountered in practice. Therefore, we assume in the sequel that $\min\{\alpha(\theta), \beta(\rho)\} > 0$. Note that this always holds if $p_\theta$ and $p_\rho$ are set via (8) and (9).

An important difference between the original LS model and the one studied here is that, for reasons which become clear in the sequel, we assume that the probability of recombination is *constant* between each site. The same model was also recently considered by [9], and is appropriate for large haplotype panels where the marker density is high and relatively uniform. It would not necessarily be appropriate for small data sets typed at a sparse set of markers.

Equation (5) asserts that log-likelihood of LS given $\pi$ is, up to an irrelevant constant, simply a weighted combination of the number of template switches and sequence mismatches. Naturally, the weights depend on the mutation and recombination parameters, with higher values of $\theta$ (resp. $\rho$) leading to lower values of $\alpha(\theta)$ (resp. $\beta(\rho)$), and correspondingly less weight placed on mismatches (resp. recombinations).

### Calculating all possible haploid decodings

In this section we derive an algorithm partition $(h)$ to efficiently calculate all possible haploid decodings for various settings of $\theta$ and $\rho$. That is, for a given focal haplotype $h$, partition $(h)$ returns a partition $S_1, \ldots, S_K$ such that

$$\bigcup_{k=1}^{K} S_k = \{(\theta, \rho) : \min\{\alpha(\theta), \beta(\rho)\} > 0\}$$

and for any $i$ and $(\theta, \rho), (\theta', \rho') \in S_i$,

$$\mathrm{LS}_h(\theta, \rho) = \mathrm{LS}_h(\theta', \rho').$$

Note that there can be multiple *paths* that achieve the optimal cost $\mathrm{LS}_h(\theta, \rho)$; the regions returned by

partition ($h$) have the property that the cost of any such path is the same within each region.

We arrive at the algorithm by a series of reductions. The first trivial result reminds us that, although LS is technically a two-parameter model, any choice of $(\theta, \rho)$ lies on a one-dimensional manifold of equivalent solutions.

**Lemma 1** *Let $c = \beta(\rho)/\alpha(\theta)$. Then for any $\theta', \rho'$ such that*

$$p_{\rho'} = \frac{1}{N + \left(\frac{p_{\theta'}}{1-3p_{\theta'}}\right)^{-c}}, \tag{10}$$

*we have* $\mathrm{LS}_h(\theta', \rho') = \mathrm{LS}_h(\theta, \rho)$.

***Proof*** If $\rho'$ and $\theta'$ satisfy (10), then $\beta(\rho')/\alpha(\theta') = c$. Hence, by Eq. (5),

$$\begin{aligned}
\mathrm{LS}_h(\theta, \rho) &= \max_{\pi} \log p(h \mid \pi, \theta, \rho) \\
&= \min_{\pi} m(\pi) + ck(\pi) \\
&= \min_{\pi} \alpha(\theta')m(\pi) + \beta(\rho')k(\pi) \\
&= \mathrm{LS}_h(\theta', \rho').
\end{aligned}$$

□

By the preceding result, we may assume that $\alpha(\theta) = 1$ in Eq. (5). Define the optimal value function

$$V_\ell(\beta) = \min_{\pi \in [N]^\ell} m(\pi) + \beta k(\pi), \tag{11}$$

so that the output of LS for a given $\beta$ is $V_L(\beta)$. All possible outputs of LS are thus contained in the set

$$\{(\beta, V_L(\beta)) : \beta \geq 0\}. \tag{12}$$

To compute this set we proceed recursively. First, define

$$V_\ell(n; \beta) \stackrel{\text{def}}{=} \min_{\substack{\pi \in [N]^\ell \\ \pi_\ell = n}} m(\pi) + \beta k(\pi)$$

to be the optimal $\ell$-path which copies from haplotype $n \in [N]$ at the terminal position. Thus,

$$V_\ell(\beta) = \min_{n \in [N]} V_\ell(n; \beta). \tag{13}$$

Plugging the definitions of $m(\pi)$ and $k(\pi)$ (Eqs. 1 and 2 ), we obtain the recurrence

$$\begin{aligned}
V_{\ell+1}(n; \beta) &= \min_{\substack{\pi \in [N]^{\ell+1} \\ \pi_{\ell+1} = n}} m(\pi) + \beta k(\pi) \\
&= \min_{\substack{\pi \in [N]^{\ell+1} \\ \pi_{\ell+1} = n}} d_{\ell+1}(n) + m(\pi_{1:\ell}) + \beta \mathbf{1}_{\{\pi_\ell \neq n\}} + \beta k(\pi_{1:\ell}) \\
&= d_{\ell+1}(n) + \min\{V_\ell(n; \beta), V_\ell(\beta) + \beta\},
\end{aligned} \tag{14}$$

where $d_\ell(n)$ is an indicator function that whether there is a copying error from haplotype $n$ at the terminal position $\ell + 1$. It is easy to see that the functions $V_\ell(n; \beta)$ and $V_\ell(\beta)$ are piecewise linear and concave in $\beta$. Hence, dynamic programming can be used to solve (11) for all values of $\beta$, repeatedly applying (14) to determine the correct piecewise representation for $V_\ell(\beta)$. Repeating this procedure for $\ell = 1, \ldots, L$, we eventually arrive at the piecewise-defined $V_L(\beta)$, i.e. Eq. (12).

We experimented with this approach but found it to be too slow in practice. Eqs. (13) and (14) require taking the pointwise minimum of a collection of $N$ piecewise linear functions. This entails finding all their points of intersection, which, though conceptually straightforward, is computationally burdensome for large $N$.

Instead, we derive an alternative algorithm that uses convex analysis to efficiently calculate $V_L(\beta)$. The algorithm recurses on a different quantity

$$J_\ell(r) \stackrel{\text{def}}{=} \min_{\substack{\pi \in [N]^\ell \\ k(\pi) = r}} m(\pi), \tag{15}$$

which is the least number of mismatches among all $(\ell, r)$ -paths. We then use a theorem from the changepoint detection literature to relate $V_\ell(\beta)$ and $J_\ell(r)$.

The theorem and ensuing discussion rely on the following basic results and definitions from convex analysis. A set $K$ is *convex* if for all $x, y \in K$, the line $[x, y] \stackrel{\text{def}}{=} \{\alpha x + (1-\alpha)y : \alpha \in [0, 1]\} \subset K$. A point $x \in K$ is a *vertex* if, for all $y, z \in K$ such that $x \in [y, z]$ (the line segment from $y$ to $z$), either $x = y$ or $x = z$. Given a set $X$, the *convex hull of $X$* is the intersection of all convex sets that contain $X$. If $X \subset \mathbb{R}^2$ and $|X| < \infty$, the convex hull of $X$ is a polygon, and can be completely described by the locations of its vertices. We use the notation

$\mathrm{conv}(X)$ to denote the convex hull of a finite set $X$ in the plane, and $\mathrm{vtx}(X)$ to denote the vertices of its convex hull.

The following key result is due to [10]. We state it in an adapted form, and provide a short proof for completeness.

**Theorem 2** ([10]) *Let*

$$\mathcal{J}_\ell = \{(r, J_\ell(r)) : r \in [\ell - 1]\} \tag{16}$$

*be the graph of $J_\ell$, and let $r_1 < \cdots < r_M$ be such that*

$$\mathrm{vtx}(\mathcal{J}_\ell) = \{(r_1, J_\ell(r_1)), \ldots, (r_M, J_\ell(r_M))\}.$$

*Then*

$$V_\ell(\beta) = \min_{r_i} J_\ell(r_i) + \beta r_i, \quad \beta_i \le \beta \le \beta_{i+1},$$

*where*

$$\beta_i = \frac{J_\ell(r_i) - J_\ell(r_{i+1})}{r_{i+1} - r_i}.$$

***Proof*** Since

$$J_\ell(r) + \beta r = \min_{\substack{\pi \in [N]^\ell \\ k(\pi) = r}} m(\pi) + \beta r = \min_{\substack{\pi \in [N]^\ell \\ k(\pi) = r}} m(\pi) + \beta k(\pi),$$

we have that

$$V_\ell(\beta) = \min_r J_\ell(r) + \beta r \tag{17}$$

is the pointwise minimum of a collection of functions which are linear in $\beta$. Thus, there exists points $r_1 < \cdots < r_M$ such that $V_\ell(\beta)$ is piecewise linear, with vertices $\beta_i$ that satisfy

$$J_\ell(r_i) + \beta_i r_i = J_\ell(r_{i+1}) + \beta_i r_{i+1}.$$

At each such $r_i$, (17) implies that

$$\max_{r < r_i} \frac{J_\ell(r_i) - J_\ell(r)}{r - r_i} < \beta_i < \min_{r > r_i} \frac{J_\ell(r) - J_\ell(r_i)}{r_i - r}. \tag{18}$$

The preceding display establishes that $(r_i, J_\ell(r_i))$ cannot be written as a convex combination of any two other points in $\mathcal{J}_\ell$, so it is a vertex of $\mathrm{conv}(\mathcal{J}_\ell)$. □

By Theorem 2, determining $V_\ell(\beta)$ reduces to finding convex hull of the graph of $J_\ell(r)$. Now let

$$J_\ell^{(n)}(r) = \min_{\substack{\pi \in [N]^\ell \\ k(\pi) = r \\ \pi_\ell = n}} m(\pi) \tag{19}$$

be the minimal number of mismatches among all $(\ell, r, n)$ -paths, and let

$$\mathcal{J}_\ell^{(n)} = \{(r, J_\ell^{(n)}(r)) : r \in [\ell - 1]\} \tag{20}$$

be its graph. We call an $(\ell, r, n)$-path $\pi$ *locally active* if $(r, m(\pi)) \in \mathrm{vtx}(\mathcal{J}_\ell^{(n)})$. Similarly, an $(\ell, r)$-path $\pi$ is *(globally) active* if $(r, m(\pi)) \in \mathrm{vtx}(\mathcal{J}_\ell)$.

By the preceding discussion, the set of active $\ell$-paths completely characterizes $V_\ell(\beta)$. The next result establishes that this set in turn may be obtained from the locally active $\ell$-paths. Let

$$\tilde{\mathcal{J}}_\ell^{(n)} = \{(r, y) : r \in [\ell - 1], y \in [J_\ell^{(n)}(r), \ell] \cap \mathbb{Z}\}$$
$$\tilde{\mathcal{J}}_\ell = \{(r, y) : r \in [\ell - 1], y \in [J_\ell(r), \ell] \cap \mathbb{Z}\}$$

be the "truncated epigraphs" of $J_\ell$ and $J_\ell^{(n)}$, comprising all of the lattice points between the corresponding sets and the line $y = \ell$. These sets have the same upper boundary and obviously $(0, \ell)$ and $(\ell - 1, \ell)$ are two common extreme points. Next, we characterize the extreme points of $\bigcup_{n=1}^N \tilde{\mathcal{J}}_\ell^{(n)}$:

**Lemma 3** *Let $A = \bigcup_{n=1}^N \tilde{\mathcal{J}}_\ell^{(n)}$ and*

$$B = \left\{(r, \min_n J_\ell^{(n)}(r)) : r \in [\ell - 1]\right\} \cup \{(0, \ell), (\ell - 1, \ell)\}.$$

*Then* $\mathrm{vtx}(A) \subset B$.

***Proof*** We have $B \subset A$, so let $(r, y) \in A \setminus B$. Then either:

1. $y \notin \{\ell, \min_n J_\ell^{(n)}(r)\}$, so that $(r, y)$ can be written as the linear combination of $(r, \ell)$ and $(r, \min_n J_\ell^{(n)}(r))$; or
2. $y = \ell$ and $r \notin \{0, \ell - 1\}$, so that $(r, y)$ is the linear combination of $(0, \ell)$ and $(\ell - 1, \ell)$.

This shows that $(r, \ell) \notin B \implies (r, \ell) \notin \mathrm{vtx}(A)$, which is equivalent to the claim. □

The following foundational result in convex analysis is stated for reference:

**Theorem** (Krein–Milman) *If $K \subset \mathbb{R}^d$ is compact and convex, then $K = \mathrm{conv}(\mathrm{vtx}(K))$.*

Since every set considered here is a finite set in $\mathbb{R}^2$, the Krein-Milman theorem always applies.

**Proposition 1** $\mathrm{vtx}(\tilde{\mathcal{J}}_\ell) = \mathrm{vtx}\left(\bigcup_{n=1}^N \tilde{\mathcal{J}}_\ell^{(n)}\right)$.

***Proof*** Since $\bigcup_{n=1}^{N} \tilde{\mathcal{J}}_\ell^{(n)}$ contains finitely many points, by the Krein-Milman theorem, its convex hull is spanned by its extreme points. Now by Lemma 3, the extreme points of $\bigcup_{n=1}^{N} \tilde{\mathcal{J}}_\ell^{(n)}$ is a subset of

$$\{(r, \min_n J_\ell^{(n)}(r)) : r \in [\ell - 1]\} \cup \{(0, \ell), (\ell - 1, \ell)\}$$

which is contained in $\tilde{\mathcal{J}}_\ell$ by definition. Thus, $\mathrm{vtx}(\bigcup_{n=1}^{N} \tilde{\mathcal{J}}_\ell^{(n)}) \subset \mathrm{vtx}(\tilde{\mathcal{J}}_\ell)$. The other direction is by noticing $\tilde{\mathcal{J}}_\ell \subset \bigcup_{n=1}^{N} \tilde{\mathcal{J}}_\ell^{(n)}$. □

At this point, we have reduced the original problem of determining $V_L(\beta)$ to that of finding the set of locally active $(L, n)$ paths for $n = 1, \ldots, N$. The next and final result shows how to compute these sets recursively. In theorem, we use an additional bit of notation: if $\pi$ is an $\ell$-path, and $c \in [N]$, then we write $\pi c$ to denote an "extension" $(\ell + 1)$-path, such that $(\pi c)_i = \pi_i$ for $i = 1, \ldots, \ell$, and $(\pi c)_{\ell+1} = c$.

**Proposition 2** *Let $\pi = \phi n$. If $\pi$ is a locally active $(\ell + 1, r, n)$-path, then either a) $\phi$ is a locally active $(\ell, r, n)$-path, or b) $\phi$ is an active $(\ell, r - 1)$-path.*

***Proof*** First suppose that $\phi_\ell = n$. We claim that $\phi$ must be locally active. If not, then there exists a locally active $(\ell, r_1, n)$ path $\phi^{(1)}$, a locally active $(\ell, r_2, n)$ path $\phi^{(2)}$, and a number $\alpha \in (0, 1)$, such that $r = \alpha r_1 + (1 - \alpha) r_2$ and

$$\alpha m(\phi^{(1)}) + (1 - \alpha) m(\phi^{(2)}) < m(\phi). \tag{21}$$

Adding $d_{\ell+1,n} = \alpha d_{\ell+1,n} + (1 - \alpha) d_{\ell+1,n}$ to both sides, we obtain

$$\alpha m(\phi^{(1)} n) + (1 - \alpha) m(\phi^{(2)} n) < m(\phi n) = m(\pi), \tag{22}$$

contradicting the fact that $\pi$ is locally active.

Next, suppose that $\phi_\ell \neq n$. Then, since $\pi$ is an $(\ell + 1, r)$ -path, $\phi$ is an $(\ell, r - 1)$-path. If $\phi$ is not active, then one may similarly find active $(\ell, r_1)$ and $(\ell, r_2)$ paths $\phi^{(1)}$ and $\phi^{(2)}$ such that inequality (21) holds, where now $r - 1 = \alpha r_1 + (1 - \alpha) r_2$. Assuming without loss of generality that $r_1 < r_2$, this implies $r_1 < r - 1 < r_2$. Path $\phi_1$ may be extended to the $(\ell + 1, r_1 + 1, n)$-path $\phi^{(1)} n$, and similarly for $\phi^{(2)}$, whence (22) holds. Because $r_1 + 1 < r < r_2 + 1$, we have $\phi^{(i)} n \neq \pi$ for $i = 1, 2$. Thus, $\pi$ is an interior point in the convex hull of all $(\ell + 1, n)$ paths, so it is not locally active. Hence, in either case we arrive at a contradiction. □

**Algorithm 1** Haploid solution surface

**Require:** Procedure VTX$(X)$ which returns the vertices of conv$(X) \subset \mathbb{R}^2$.

**procedure** PARTITION(h)
  // $\mathcal{J}_\ell, \mathcal{J}_\ell^{(n)}$ contain vertices of convex hulls of active paths.
  Initialize $\mathcal{J}_0 = \mathcal{J}_0^{(n)} = \{(0, 0)\}$ for $n = 1, \ldots, N$
  **for** $\ell = 1, \ldots, L$ **do**
    $u = \emptyset$
    **for** $n = 1, \ldots, N$ **do**
      $p_1 = \{(r, m + d_{\ell,n}) : (r, m) \in \mathcal{J}_{\ell-1}^{(n)}\}$ ▷ extensions
      $p_2 = \{(r + 1, m + d_{\ell,n}) : (r, m) \in \mathcal{J}_{\ell-1}\}$ ▷ recombinations
      $\mathcal{J}_\ell^{(n)} = \mathrm{VTX}(p_1 \cup p_2)$
      $u = \mathrm{VTX}(u \cup \mathcal{J}_\ell^{(n)})$
    **end for**
    $\mathcal{J}_\ell = \mathrm{VTX}(u)$
  **end for**
  **return** $\mathcal{J}_L$ ▷ Active paths at position $L$
**end procedure**

By the preceding results, in order to find the set of active $(\ell+1)$-paths, it is only necessary to keep track of the set of active $\ell$-paths, as well as the set of locally active $(\ell, n)$-paths for each haplotype $n \in [N]$. Algorithm 1 implements Proposition 2. The output of the algorithm is $\mathrm{vtx}(\mathcal{J}_L)$. From this, Theorem 2 can be used to calculate $\infty = \beta_0 > \beta_1 > \cdots$ such that $\mathrm{ls}_h(1, \beta)$ is constant on each interval $\beta \in (\beta_i, \beta_{i-1})$. Finally, Lemma 1 and Eqs. (8)–(9) yield the solution space for all $(\theta, \rho)$.

A few implementation details of Algorithm 1 are worth mentioning. As can be seen from lines 7–8, the assumption that $\alpha(\theta) \equiv 1$ causes the locally and globally active vertices to live on the lattice: $\mathcal{J}_\ell, \mathcal{J}_\ell^{(n)} \in \mathbb{Z}^2$. All numerical calculations are therefore exact, so the algorithm is impervious to rounding errors, or other floating point concerns. Also, for a finite set $X \subset \mathbb{R}^2$, and assuming that the points in $X$ are already sorted by their $x$-coordinates, the operation $\mathrm{conv}(X)$ used in lines 9 and 10 can be carried out in $O(|X|)$ operations using e.g. Andrew's algorithm [11]. This can easily be achieved by storing $\mathcal{J}_\ell^{(n)}$ and $\mathcal{J}_\ell$ as sorted linked lists, and appropriately merging them in lines 9–10 instead of performing a naive set union. As the output of Andrews' algorithm remains sorted, this ensures that the number of operations needed to perform lines 7–10 is minimized for all $\ell$. It should be noted that, in practice, these optimizations may not improve performance unless $L$ and $N$ are very large. Finally, lines 7–9 are embarrassingly parallel and can be performed simultaneously using $N$ different threads. However, the final reduction step (line 10) requires synchronization.

#### The diploid algorithm

The diploid extension to the Li-Stephens algorithm [e.g., 12, 13] finds a pair of copying paths $(\pi_1, \pi_2) \in [N]^{2\times L}$ that maximizes the probability of observing a sequence of diploid genotypes $g \in \mathcal{D}^{2\times L}$. Similar to the haploid case, the log-likelihood of $g$ given $(\pi_1, \pi_2)$ has a compact expression [9]:

$$-\log p(g \mid \theta, \rho) = \alpha(\theta) m(\pi_1, \pi_2) + \beta(\rho)[r(\pi_1) + r(\pi_2)] + C, \tag{23}$$

where $m(\pi_1, \pi_2)$ was defined in Eq. 3.

We define $\mathrm{LS}_g(\theta, \rho)$ analogously to return a path pair $(\pi_1^*, \pi_2^*)$ which minimizes Eq. (23). Clearly, Lemma 1 goes through without modification for $\mathrm{LS}_g(\theta, \rho)$ as well, so it is only necessary to determine the solution path for $\mathrm{LS}_g(1, \rho)$. Algorithm 2 does this. The idea of the algorithm is similar to the haploid case, however more work is required in the form of an additional inner `for` loop needed to track both single and double recombination events. For each $n_1, n_2 \in [N]$, the algorithm tracks a new set $\mathcal{J}_\ell^{(n_1,n_2)}$ of locally active path *pairs*, as well as sets $\mathcal{J}_\ell^{(n_1)}$ of "partially active" paths which lie on the convex hull of path costs involving haplotype $n_1$ only. The set of "active" paths $\mathcal{J}_\ell$ is now the convex hull of path costs taken over all possible path pairs.

The proof of correctness relies on a generalization of Proposition 2.

**Proposition 3** *Suppose that* $(\pi n_1, \lambda n_2)$ *is an active* $(\ell+1, r, (n_1, n_2))$*-path. Then one of the following is true*:

- $(\pi, \lambda)$ is a locally active $(\ell, r, (n_1, n_2))$ path.
- $\pi$ is a partially active $(\ell, r-1, n_1)$ path.
- $\lambda$ is a partially active $(\ell, r-1, n_2)$ path.
- $(\pi, \lambda)$ is an active $(\ell, r-2)$ path.

***Proof*** Similar to Proposition 2, the proof amounts to conditioning on last entries of $\pi$ and $\lambda$, and showing that those paths must lie on the convex hull of the appropriate set of $\ell$-paths. There are four cases to check depending on whether $\pi_\ell = n_1$ and/or $\lambda_\ell = n_2$. We prove one case and omit the repetitive details for the other three. Suppose that $\pi_\ell = n_1$ and $\lambda_\ell = n_2$, but that $(\pi, \lambda)$ is not locally active. Then there are locally active $(\ell, r, (n_1, n_2))$ paths $(\phi_1, \phi_2)$ and $(\gamma_1, \gamma_2)$ such that

$$\alpha m(\phi_1, \phi_2) + (1-\alpha) m(\gamma_1, \gamma_2) < m(\pi, \lambda).$$

Thus

$$\alpha m(\phi_1 n_1, \phi_2 n_2) + (1-\alpha) m(\gamma_1 n_1, \gamma_2 n_2) < m(\pi n_1, \lambda n_2),$$

contradicting the supposition. □

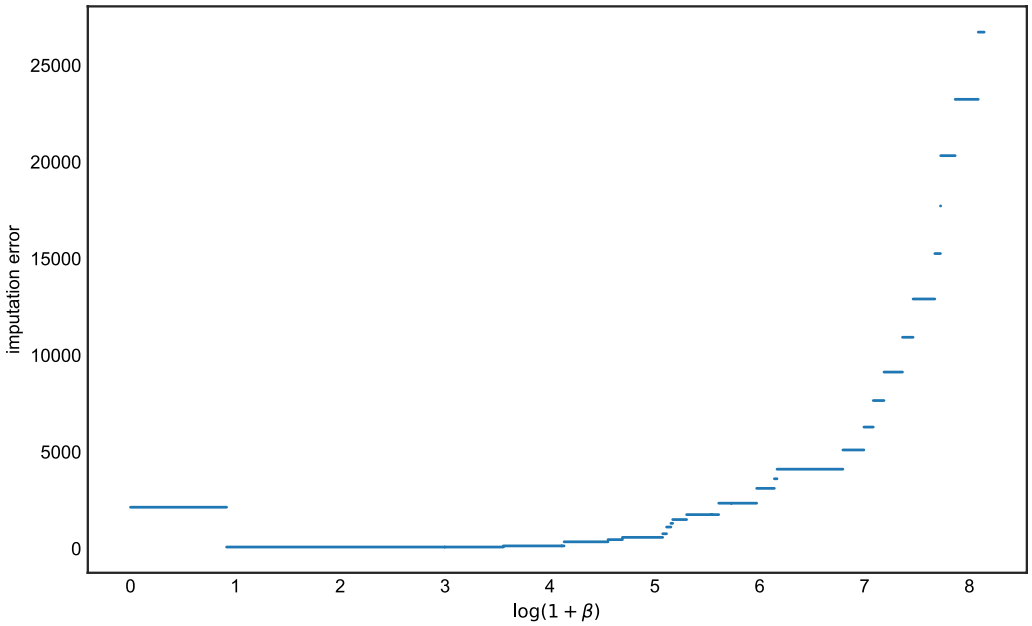


**Fig. 1** The imputation errors of all possible $\beta$ for the haploid case, the $x$ axis is the value of $\log(1+\beta)$ and the $y$ axis is the corresponding value of imputation error

**Algorithm 2** Diploid solution surface

**Require:** Procedure $\textsc{vtx}(X)$ which returns the vertices of $\text{conv}(X) \subset \mathbb{R}^2$.

**procedure** PARTITION(g)
  // $\mathcal{J}_\ell, \mathcal{J}_\ell^{(n)}, \mathcal{J}_\ell^{(n,m)}$ are vertices of active paths.
  Initialize $\mathcal{J}_0 = \mathcal{J}_0^{(n)} = \mathcal{J}_0^{(n,m)} = \{(0,0)\}$ for $n, m = 1, \ldots, N$
  **for** $\ell = 1, \ldots, L$ **do**
    **for** $n_1 = 1, \ldots, N$ **do**
      $\mathcal{J}_\ell^{(n_1)'} = \emptyset$
      **for** $n_2 = 1, \ldots, N$ **do**
        $p_1 = \{(r, m + d_{\ell,(n_1,n_2)}) : (r,m) \in \mathcal{J}_{\ell-1}^{(n_1,n_2)}\}$ ▷ extensions
        $p_2 = \{(r+1, m + d_{\ell,(n_1,n_2)}) : (r,m) \in \mathcal{J}_{\ell-1}^{(n)} \cup \mathcal{J}_{\ell-1}^{(m)}\}$ ▷ single recomb.
        $p_3 = \{(r+2, m + d_{\ell,(n_1,n_2)}) : (r,m) \in \mathcal{J}_{\ell-1}\}$ ▷ double recomb.
        $\mathcal{J}_\ell^{(n_1)'} = \textsc{vtx}(\mathcal{J}_\ell^{(n_1)'} \cup p_1 \cup p_2)$
        $\mathcal{J}_\ell^{(n_1,n_2)} = \textsc{vtx}(p_1 \cup p_2 \cup p_3)$
      **end for**
    **end for**
    $J_\ell = \emptyset$
    **for** $n = 1, \ldots, N$ **do**
      $\mathcal{J}_\ell^{(n)} = \mathcal{J}_\ell^{(n)'}$
      $\mathcal{J}_\ell = \textsc{vtx}(\mathcal{J}_\ell, \mathcal{J}_\ell^{(n)})$
    **end for**
  **end for**
  **return** $\mathcal{J}_L$ ▷ Vertices of convex hull
**end procedure**

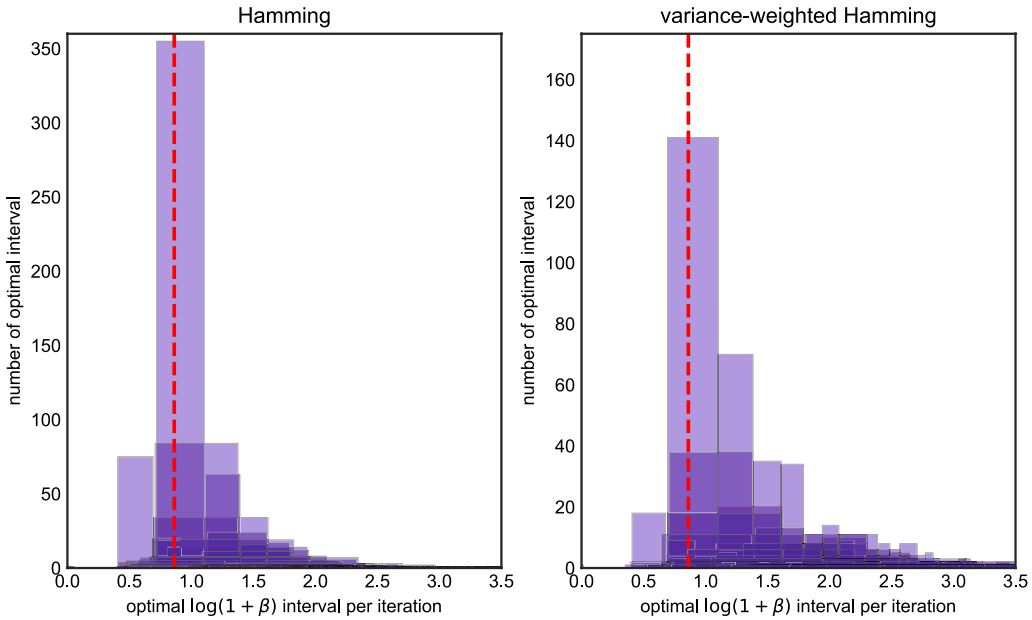


**Fig. 2** The histogram of optimal $\beta$ intervals for Algorithm 1, the *x* axis is the value of optimal $\log(1+\beta)$ intervals in each iteration, the *y* axis is the number of replicates in 1000 iterations. The *x* axis of the red dash line is the true value of $\log(1+\beta_0)$ we used to generate data. The left panel is the histogram under the Hamming loss, the right panel is histogram under the weighted Hamming loss

## Results

We used our algorithm to study two of the main use cases for LS: phasing and imputation. The phasing problem attempts to resolve a sequence of diploid genotypes $g \in \mathcal{D}^{2\times L}$ into a pair of haplotypes $h_1, h_2 \in \mathcal{D}^L$, such that $g$ and $h_1, h_2$ possess the same alleles at each position, and switching error is minimized. In the imputation problem, missing positions in a single haplotype are imputed using data from a reference panel. Notable phasing and imputation algorithms based on the LS model include fastPHASE [14], IMPUTE2 [15], MaCH [16], SHAPEIT [17], and EAGLE [13].

### *Investigating imputation accuracy using Algorithm 1*

For testing the Algorithm 1, we consider a haplotype imputation problem. Given a haplotype with the information of some SNPs is missing, we impute the haplotype with all possible $\beta$ using the algorithm. To study imputation error, we considered the loss function

$$\sum_{\ell=1}^{L} \omega_\ell |X_\ell^{\text{true}} - X_\ell^{\text{imp}}|,$$

where the $\omega_i$ are position-specific weights. We considered two choices for the weights: $\omega_i \equiv 1$, corresponding to Hamming distance between the imputed and true haplotypes; and $\omega_\ell = [\text{MAF}_\ell(1-\text{MAF}_\ell)]^{-1}$, where $\text{MAF}_\ell$ is the minor allele frequency at position $\ell$, thereby upweighting rare variants in the loss calculation.

The way we ran our algorithm is as follows: we first generate a focal haplotype $h$ and reference panel $H$. The focal haplotype is then chosen as the underlying truth, and then all loci with minor allele frequency (MAF) less than 0.05 are discarded. We then use the retained loci to compute the solution surface, i.e. for a sequence of $\beta$, we compute the corresponding optimal path $p = \{\pi_{p_1}, \ldots, \pi_{p_k}\}$ with length $k$ for each $\beta$. A missing locus is imputed by pasting the copying path state from the nearest flanking marker. The number of mismatches between the imputed copying path and the truth is computed in the end.

We simulated 1001 sequences with 100Mb in a single population using `msprime` [18, 19]. The length was chosen to be comparable to the size of a typical human chromosome. The effective population size was fixed to 1, and the scaled rates of recombination and mutation were both set to be $10^{-4}$ per unit of coalescent time. This resulted in a binary genotype matrix with roughly 300,000 rows and 1001 columns. For the haplotype imputation, we used the first column as the focal haplotype to be imputed, and the remaining columns 2–1001 as the haplotype panel. We then introduced missing data according to the MAF threshold mentioned above. This resulted in approximately 40% of the loci being retained on average.

For a given dataset, we first ran Algorithm 1 in order to find all possible LS paths. Then, for each interval of $\beta$ where the LS solution has constant cost, we chose an

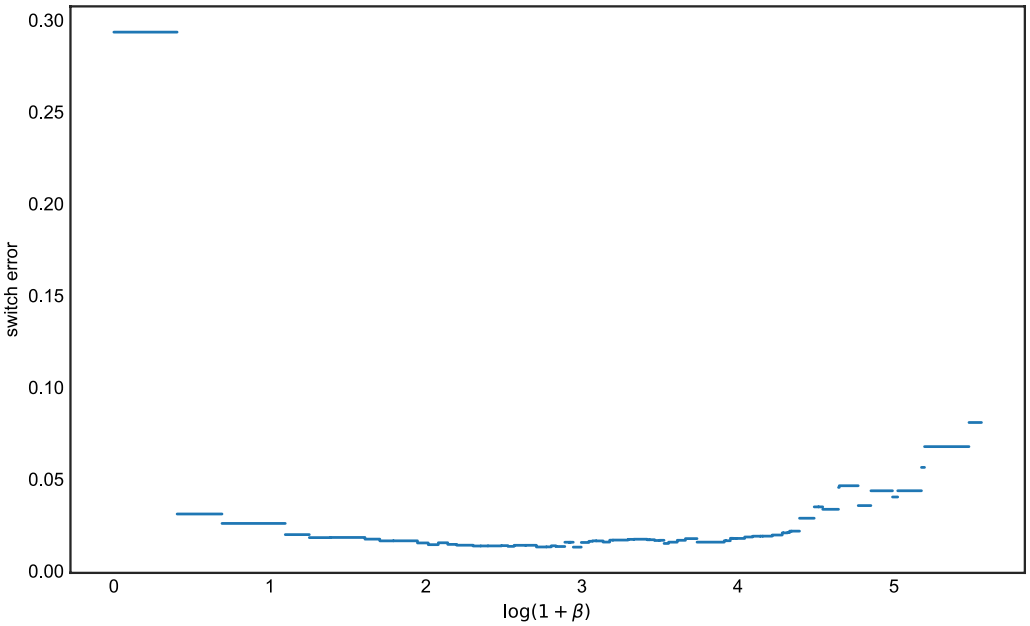


**Fig. 3** The switch errors of all possible $\beta$ for the diploid case, the $x$ axis is the value of $\log(1+\beta)$ and the $y$ axis is the corresponding value of the switch error

optimal path and recorded its imputation error.[1] Fig. 1 shows the results of a single experiment for the Hamming loss. The curve is piecewise constant, with jumps at points where increasing $\beta$ causes the cost of the optimal LS path to change. For this particular simulation, any setting $0.9 \leq \log(1+\beta) \leq 4.1$ (roughly) was optimal in terms of imputation error. At the extremes $\beta = 0$ and $\beta \to \infty$, we see the expected behavior. When $\beta = 0$ (i.e., $\rho \to \infty$ in Eq. 6), there is free recombination so a copying path that contains zero copying errors (mutations) can be achieved. However, this results in some imputation errors since LD information is no longer being used for imputation. And as $\beta \to \infty$, which implies $\rho = 0$ and complete linkage, the algorithm simply copies in entirety from the most closely related haplotype with no recombinations, resulting in many imputation errors.

We repeated this procedure 1,000 times and for each iteration, we determined the interval of $\beta$ for which imputation error was minimized. Figure 2 depicts these results. Each box in the plot represents corresponds to an interval which was optimal in at least one run, with the height of the box representing the number of times it was the optimal interval across all 1,000 runs. (Because the corners of each box are all integers, they are displayed with transparency and a small amount of jitter to reduce overplotting.) The red dashed line in the plot corresponds to setting $\beta$ according to Eq. (6) and (8), suitably transformed using Lemma 1, where $\rho$ is the population-scaled rate of recombination. If the red line lies inside an interval, it means the LS model run with the population-scaled rate of recombination has the optimal imputation error. Otherwise, the results of imputation could be improved by choosing a different setting of $\beta(\rho)$.

The left-hand panel of Fig. 2 measures error in terms of Hamming loss, whereas the right-hand panel uses (inverse) variance-weighted loss. For Hamming loss, most of the optimal intervals are contained in $\log(1+\beta) \in [0.5, 2.0]$, and parameterizing LS using the population-scaled values for $\theta$ and $\rho$ generally falls inside the optimal interval (in roughly 57% of the runs). The right-hand panel shows most of the optimal intervals lie are contained in between $\log(1+\beta) \in [0.75, 1.4)$. The same is mostly true for variance-weighted loss, however there is a longer right-tail to the distribution, and some simulations where setting $\beta$ much larger (around 2.5–3) could lower imputation error. The percentage that the population-scaled values for $\theta$ and $\rho$ falls inside the optimal interval is 0.246.

*Simulation study with variable effective population size* Next, we considered a more complex scenario where the population size varied according to a realistic model of human history. We simulated data using `stdpopsim` [20], using the `Africa_1T12` demography for

[1] Note that, while multiple copying paths may have the same error according to Eq. (5), they may have different imputation errors. We verified that the results presented below were consistent from run to run, and not driven by arbitrary choices of the optimal copying path.

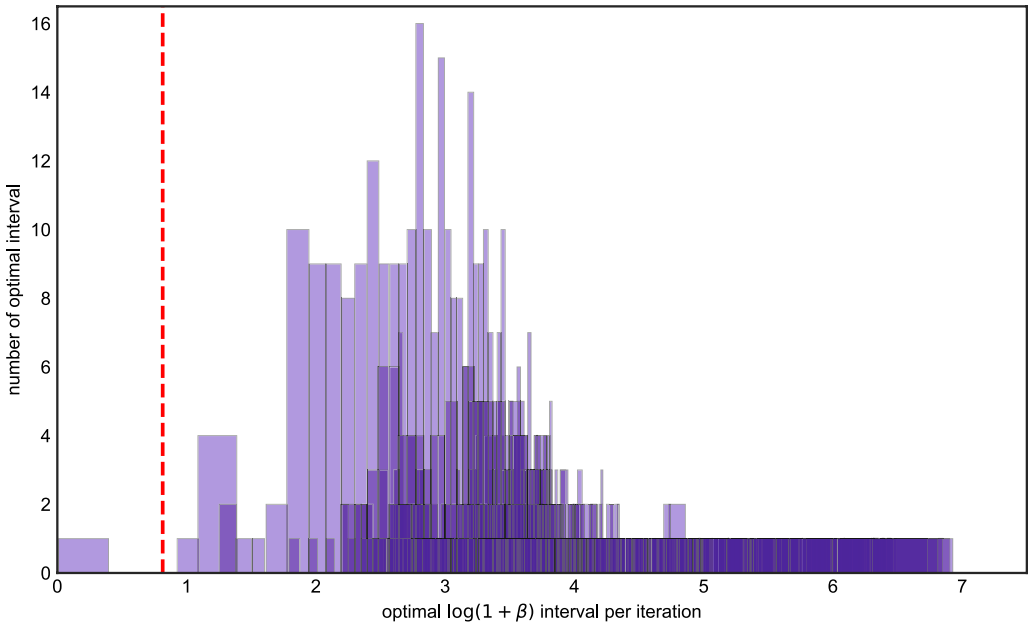


**Fig. 4** The histogram of optimal $\beta$ intervals for Algorithm 2, the *x* axis is the value of optimal $\log(1+\beta)$ intervals in each iteration, the *y* axis is the number of replicates in 1000 iterations. The *x* axis of the red dash line is the true value of $\log(1+\beta_0)$ we used to generate data

*H. sapiens*. This is a simplified two-population model with the European-American population being removed, and it describes the ancestral African population together with the out-of-Africa event [21]. We simulated 1001 samples of human chromosome 2, but artificially reduced the length of the chromosome to be around 100Mb for computational reasons and to match the preceding experiment. The scaled mutation rate and recombination rate were around $8.7 \times 10^{-4}$ and $9.8 \times 10^{-4}$ respectively. We set the first sample as the focal and 2 to 1001 samples as the panel. The imputation procedure was the same as in the preceding section, where we retained the loci with MAF $> 0.05$ and imputed the remaining sites. To determine the population-scaled mutation rate, we set $N_e = \mathbb{E}T_{\text{MRCA}}/2$, where $\mathbb{E}T_{\text{MRCA}}$ is the average time to coalescence in a sample of two chromosomes under the `Africa_1T12` demography, and then and then defined $\theta = 4N_e\mu$.

For Hamming loss, Fig. 7 shows the distribution of the optimal intervals is less dispersed than in the fixed population size case, with optimal $\log(1+\beta)$ intervals concentrated between 0.75 and 1.2, which closely coincides with the population-scaled value (dashed red line). The percentage of runs where $\beta(\rho)$ was optimal increased, to 0.73. Only a small amount of optimal intervals fall to the left of $\beta(\rho)$. This indicates very occasionally, LS will perform better if the recombination rate is set lower than population-scaled value.

For the variance-weighted loss (Fig. 7 right panel), we observed a similar phenomenon as in the constant-size case: there is a heavier right tail, and in a larger fraction of the simulations, imputation results could have been improved by setting $\beta$ higher than the population-scaled value. The percentage of runs where $\beta(\rho)$ was optimal decreased, to 0.223. However, in general, the previous two experiments show that the population-scaled rates should generally be adequate for phasing using the haploid LS algorithm.

*Accuracy of pre-phased imputation* A common workflow for imputing diploid genotypes is to first phase them into haplotypes and then run haplotype imputation [22]. We repeated the preceding experiments to study diploid imputation using pre-phasing (Additional file 1: Figs. S1–S4) and observed generally comparable results: the population-scaled values generally result in optimal performance for pre-phasing when considering Hamming loss, but there is a longer right tail when rare variants are upweighted in the loss calculation. For the more realistic, out-of-Africa demography, we observed less dispersion of the optimal intervals than for the constant demography.

#### *Investigating phasing accuracy using Algorithm 2*

To test Algorithm 2, we considered a genotype phasing problem. Given a genotype sequence which is the pairwise sum of two haplotype sequences, and a reference panel, we aim to recover the information of each haplotype sequence. We first generated two focal haplotype

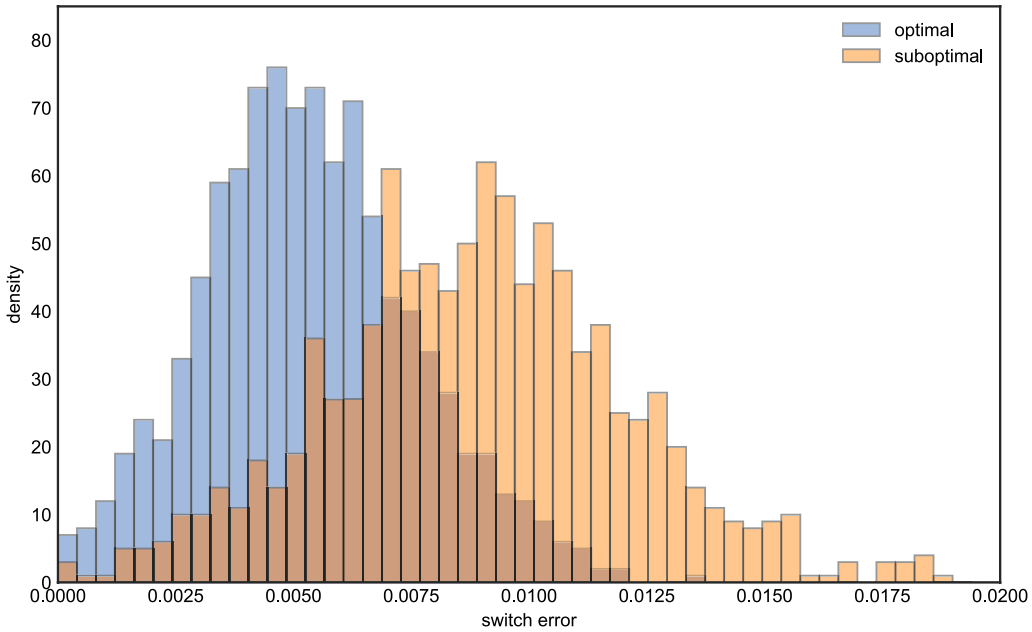


**Fig. 5** The histogram of switch errors for optimal and suboptimal $\beta$s respectively, the *x* axis is the value of switch error, the *y* axis is the density

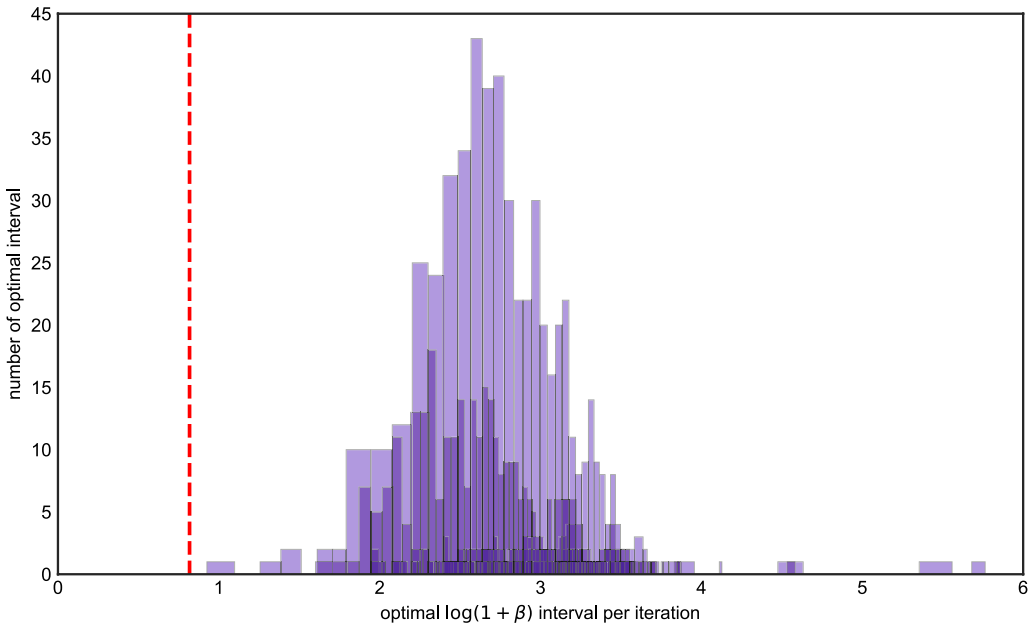


**Fig. 6** The histogram of optimal $\beta$ intervals for Algorithm 2 where the model is `Africa_1T12`, the *x* axis is the value of optimal $\log(1+\beta)$ intervals in each iteration, and the *y* axis is the number of replicates in 1000 iterations. The *x* axis of the red dash line is the value of $\log(1+\beta_0)$ where $\beta_0$ is the truth

sequences $h_1, h_2$ and reference panel $H$ and combine $h_1$ and $h_2$ to form a genotype sequence. In order to get the phased haplotype sequences, we then fed the genotype sequence and reference panel to algorithm 2. We then measured the accuracy by measuring the switch error between the true and estimated haplotype sequences. Switch error was computed using the `--diff-switch-error` option of `vcftools` [23].

We again start with the simple simulation scenario where the effective population size is fixed and equal

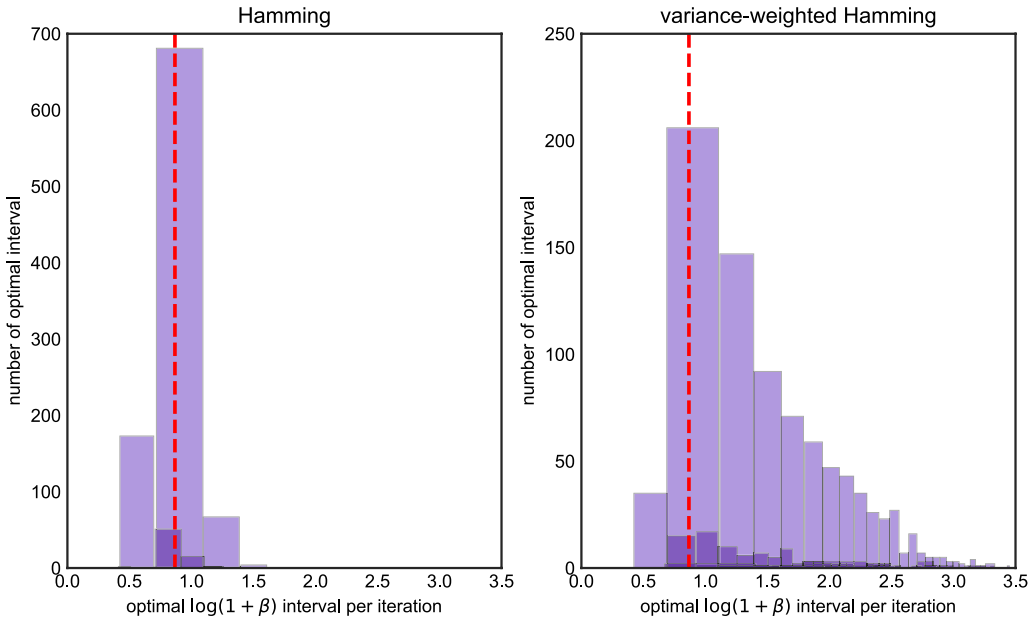


**Fig. 7** The histogram of optimal $\beta$ intervals for Algorithm 1 where the model is `Africa_1T12`, the *x* axis is the value of optimal $\log(1+\beta)$ intervals in each iteration, the *y* axis is the number of replicates in 1000 iterations. The *x* axis of the red dash line is the true value of $\log(1+\beta_0)$ we used to generate data. The left panel is the histogram under the Hamming loss, the right panel is histogram under the weighted Hamming loss

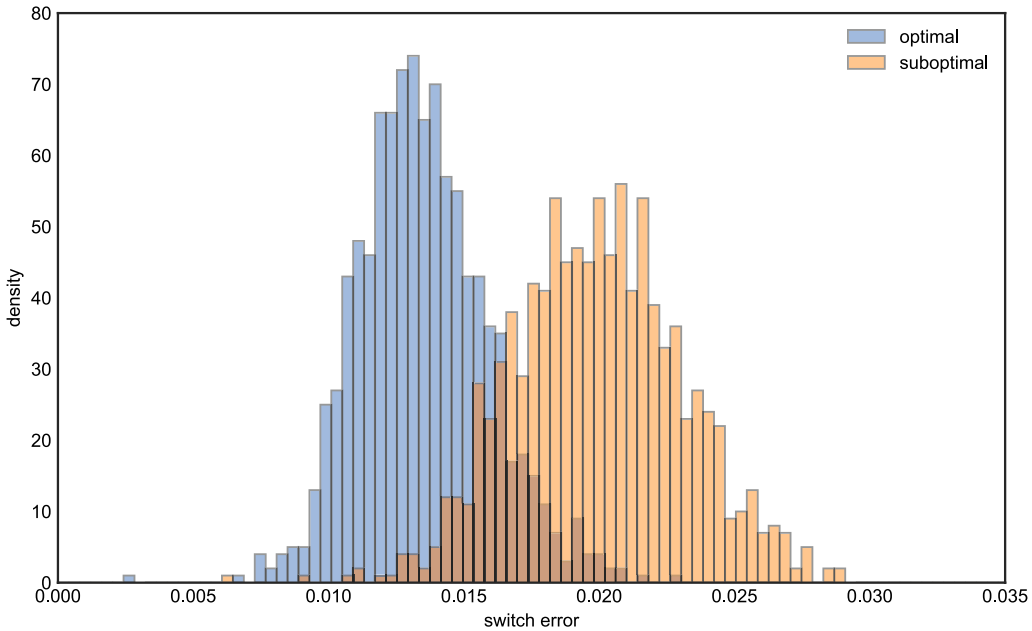


**Fig. 8** The histogram of switch errors for optimal and suboptimal $\beta$s respectively under the `Africa_1T12` model, the *x* axis is the value of switch error, the *y* axis is the density

to 1. The value of mutation and recombination rate per time are both set to be $10^{-4}$. As noted above, the diploid solution surface algorithm scales quadratically in the reference panel size, as opposed to linearly for the haploid algorithm. For this reason, we considered a shorter sequence length and a smaller panel size. We set the sequence length to be about 10MB and use a reference panel 100 haplotypes. After generating 102 samples from the model, we choose the first two columns of the genotype matrix as the focal and 3 to 102 columns as the

panel. An example of one run of the experiment is shown in Fig. 3. In this experiment, the algorithm achieves minimum switch error when $\log(1+\beta)$ is around 2.

Figure 4 shows the results of running this experiment 1000 times. In contrast to the haploid case, the optimal setting of $\beta$ is systematically higher compared to the value based on the population-scaled rate, which is again shown as a red dashed line. Although the best $\beta$ intervals seem to be more dispersed than the ones in the imputation problem, most $\log(1+\beta)$ intervals are clustered on the right of the red dash line and are between 2 and 4. Moreover, only a few $\beta$s fall into the same partition as the red line. We also noticed for that some iterations, the optimal $\beta$ interval is near 0. This can occur if there is one or more very closely related haplotypes in the reference panel. To validate these results, we compared the distribution of switch error using a $\beta$ from the modal interval in Fig. 4 (we chose 23.0), and compared it to that obtained when the $\beta$ was suboptimally set according to the scaled rate of recombination. Figure 5 shows there is a difference of switch errors by using these two $\beta$s, with the distribution of switch error under the optimal setting possessing more mass at zero.

Finally, we repeated the phasing experiment using the `Africa_1T12` demographic model. We simulated 20% of chromosome 22, so that the sequence length is about 10Mb, and imputed a diploid genotype sequence using a reference panel of size 100. From Fig. 6 we conclude the distribution of optimal $\log(1+\beta)$ intervals has a similar pattern as in the fixed population size case. Increasing $\beta$—that is, penalizing recombinations more heavily—leads to lower switch error. Figures 7 and 8 shows the differences if we use optimal and suboptimal $\beta$ respectively. The differences are more pronounced compared to the preceding section: more than half of the simulations using the "optimal" setting had lower switch error than almost every simulation using the "suboptimal" setting.

## Conclusions

In this paper, we derived a new algorithm for computing all possible solutions to the Li-Stephens haplotype copying model, as well as its diploid extension, as a function of the recombination rate parameter. Our results work by exploiting convex structure in the Viterbi decoding algorithm used to compute the optimal (frequentist) LS haplotype copying path. Our algorithms partition the LS parameter space into regions where the output of the model is constant. We showed how these can be useful for studying imputation and phasing accuracy, two of the most important uses of the LS model.

Our methods work by interpreting the LS model as a method for performing changepoint detection. Although this perspective appears to be new as far as the LS model goes (but see [24]), it has appeared in the literature before in other forms. The CROPS algorithm [25] is a general procedure for computing the solution space of changepoint models as a function of a penalty parameter, which could also be applied here. The main difference between our contribution and theirs is that the CROPS algorithm is iterative, requiring multiple runs of the model in order to compute the entire solution surface, whereas our algorithm requires only a single pass over the data. Figures 1 and 3 illustrate that, for investigating derived quantities such as phasing or imputation error, it seems necessary to compute the entire solution surface, since the error curves do not posses any sort of regularity (e.g., convexity) which would allow one to know when a globally optimal solution has been found. However, for very large data sets, the iterative approach of the CROPS algorithm may be preferable.

Some further refinements to our algorithms are possible. While we showed in simulations that for diploid phasing that there is a gap between the $\beta$ used for generating the samples and the optimal $\beta$ for LS models, our computation of $\beta$ is based on constant recombination rate. In contrast, most popular phasing and imputation packages, for example BEAGLE [26] or IMPUTE2 [15], use a recombination map whose value changes based on the genetic distance between each marker site. We do not see an easy way to modify our algorithm to accommodate this type of analysis since it is not even clear what the resulting output would be. A similar difficulty was noted by [9] in the context of the fastLS algorithm.

The size of the reference panel considered in our simulation study is relatively small, especially for the diploid phenotype phasing setting, where we only used a reference panel with a size equal to 100. We had to choose this small value because the complexity of our Algorithm 2 is at least quadratic with the size of the panel: the two nested `for` loops lead to $\mathcal{O}(N^2)$ scaling, and the number of vertices in $\mathcal{J}_\ell$ also has some dependence on $N$, though we do not currently understand the precise relationship. In contemporary imputation and phasing studies, the panel size is much larger, e.g. $2 \times 10^5$ individuals in [27]. Their study results indicate the imputation of low-frequency variants can be highly benefited from a large reference panel with accurately phased genotypes. A potential direction is thus to scale our algorithm to the setting where the size of the reference panel is large.

## Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s13015-023-00237-z.

**Additional file 1.** Additional figures.

**Acknowledgements**
The authors thank Hyun Min Kang for providing feedback on a draft of this article.

**Author contributions**
JT: Conceptualization, software, investigation, writing, supervision. YJ: Investigation, analysis, writing.

**Funding**
This research was supported by NSF grant number DMS-2052653 and NIH grant number R35GM151145.

**Availability of data and materials**
Source code implementing our method is available at https://github.com/jthlab/lsss.

## Declarations

**Ethics approval and cosnent to participate**
Not applicable.

**Competing interests**
The authors declare that they have noterests.

Received: 29 March 2023 Accepted: 30 July 2023
Published online: 09 August 2023

**References**

1. Li N, Stephens M. Modeling linkage disequilibrium and identifying recombination hotspots using single-nucleotide polymorphism data. Genetics. 2003;165:2213–33.
2. Song YS. Na Li and Matthew Stephens on modeling linkage disequilibrium. Genetics. 2016;203(3):1005–6.
3. Paul JS, Song YS. A principled approach to deriving approximate conditional sampling distributions in population genetics models with recombination. Genetics. 2010;186:321–38.
4. Efron B, Hastie T, Johnstone I, Tibshirani R. Least angle regression. Ann Stat. 2004;32(2):407–99. https://doi.org/10.1214/009053604000000067.
5. Gui J, Li H. Penalized cox regression analysis in the high-dimensional and low-sample size settings, with applications to microarray gene expression data. Bioinformatics. 2005;21(13):3001–8.
6. Sohn I, Kim J, Jung S-H, Park C. Gradient lasso for cox proportional hazards model. Bioinformatics. 2009;25(14):1775–81.
7. Huang T, Wu B, Lizardi P, Zhao H. Detection of DNA copy number alterations using penalized least squares regression. Bioinformatics. 2005;21(20):3811–7.
8. Lu Y, Zhou Y, Qu W, Deng M, Zhang C. A lasso regression model for the construction of microRNA-target regulatory networks. Bioinformatics. 2011;27(17):2406–13.
9. Lunter G. Haplotype matching in large cohorts using the Li and Stephens model. Bioinformatics. 2019;35(5):798–806.
10. Lavielle M. Using penalized contrasts for the change-point problem. Signal Process. 2005;85:1501–10. https://doi.org/10.1016/j.sigpro.2005.01.012.
11. Andrew AM. Another efficient algorithm for convex hulls in two dimensions. Inf Process Lett. 1979;9(5):216–9. https://doi.org/10.1016/0020-0190(79)90072-3.
12. Marchini J, Howie B, Myers SR, McVean G, Donnelly P. A new multipoint method for genome-wide association studies by imputation of genotypes. Nat Genet. 2007;39(7):906–13.
13. Loh P-R, Palamara PF, Price AL. Fast and accurate long-range phasing in a UK biobank cohort. Nat Genet. 2016;48(7):811.
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. 2006;78:629–44.
15. Howie BN, Donnelly P, Marchini J. A flexible and accurate genotype imputation method for the next generation of genome-wide association studies. PLoS Genet. 2009;5(6):1000529.
16. Li Y, Abecasis GR. Mach 1.0: rapid haplotype reconstruction and missing genotype inference. Am J Hum Genet. 2006;S79:2290.
17. Delaneau O, Marchini J, Zagury J-F. A linear complexity phasing method for thousands of genomes. Nat Methods. 2012;9(2):179–81.
18. Kelleher J, Etheridge AM, McVean G. Efficient coalescent simulation and genealogical analysis for large sample sizes. PLoS Comput Biol. 2016;12(5):1004842.
19. ...Baumdicker F, Bisschop G, Goldstein D, Gower G, Ragsdale AP, Tsambos G, Zhu S, Eldon B, Ellerman EC, Galloway JG, Gladstein AL, Gorjanc G, Guo B, Jeffery B, Kretzschumar WW, Lohse K, Matschiner M, Nelson D, Pope NS, Quinto-Cortés CD, Rodrigues MF, Saunack K, Sellinger T, Thornton K, van Kemenade H, Wohns AW, Wong Y, Gravel S, Kern AD, Koskela J, Ralph PL, Kelleher J. Efficient ancestry and mutation simulation with MSPRIME 1.0. Genetics. 2022;220(3):229.
20. Adrion JR, Cole CB, Dukler N, Galloway JG, Gladstein AL, Gower G, Kyriazis CC, Ragsdale AP, Tsambos G, Baumdicker F, Carlson J, Cartwright RA, Durvasula A, Kim BY, McKenzie P, Messer PW, Noskova E, Vecchyo DO-D, Racimo F, Struck TJ, Gravel S, Gutenkunst RN, Lohmeuller KE, Ralph PL, Schrider DR, Siepel A, Kelleher J, Kern AD. A community-maintained standard library of population genetic models. BioRxiv 2019; https://doi.org/10.1101/2019.12.20.885129.
21. Tennessen JA, Bigham AW, O'Connor TD, Fu W, Kenny EE, Gravel S, McGee S, Do R, Liu X, Jun G, et al. Evolution and functional impact of rare coding variation from deep sequencing of human exomes. Science. 2012;337(6090):64–9.
22. Howie B, Fuchsberger C, Stephens M, Marchini J, Abecasis GR. Fast and accurate genotype imputation in genome-wide association studies through pre-phasing. Nat Genet. 2012;44(8):955–9. https://doi.org/10.1038/ng.2354.
23. Danecek P, Auton A, Abecasis G, Albers CA, Banks E, DePristo MA, Handsaker RE, Lunter G, Marth GT, Sherry ST, McVean G, Durbin R, Group GPA. The variant call format and VCFtools. Bioinformatics. 2011;27(15):2156–8. https://doi.org/10.1093/bioinformatics/btr330.
24. Ki C, Terhorst J. Exact decoding of the sequentially Markov coalescent. J Am Stat Assoc. 2020. https://doi.org/10.1101/2020.09.21.307355.
25. Haynes K, Fearnhead P, Eckley IA. A computationally efficient nonparametric approach for changepoint detection. Stat Comput. 2017;27(5):1293–305. https://doi.org/10.1007/s11222-016-9687-5.
26. Browning BL, Tian X, Zhou Y, Browning SR. Fast two-stage phasing of large-scale sequence data. Am J Hum Genet. 2021;108(10):1880–90. https://doi.org/10.1016/j.ajhg.2021.08.005.
27. Browning BL, Browning SR. Genotype imputation with millions of reference samples. Am J Hum Genet. 2016;98(1):116–26. https://doi.org/10.1016/j.ajhg.2015.11.02.

## Publisher's Note