De novo reconstruction of microbial haplotypes by integrating statistical and physical linkage

Chen Cao[1,10], Jingni He[1,10], Lauren Mak[1,2,10], Deshan Perera[1], Devin Kwok[3], Jia Wang[4], Minghao Li[1], Tobias Mourier[5], Stefan Gavriliuc[1], Matthew Greenberg[3], A. Sorana Morrissy[1], Laura K. Sycuro[6,1], Guang Yang[1,7], Daniel C. Jeffares[8], Quan Long[1,3,7,9#]

[1]Department of Biochemistry & Molecular Biology, Alberta Children's Hospital Research Institute, University of Calgary, Calgary, Canada.

[2]Current address: Tri-Institutional Computational Biology & Medicine Program, Weill Cornell Medicine of Cornell University, NY, USA.

[3]Department of Mathematics & Statistics, University of Calgary, Calgary, Canada.

[4]Electrical and Computer Engineering, Illinois Institute of Technology, Chicago, USA.

[5]Pathogen Genomics Laboratory, Biological and Environmental Sciences and Engineering (BESE) Division, King Abdullah University of Science and Technology (KAUST), Thuwal, Saudi Arabia.

[6]Department of Microbiology, Immunology, and Infectious Diseases, Snyder Institute for Chronic Diseases, University of Calgary, Calgary, Canada.

[7]Department of Medical Genetics, University of Calgary, Calgary, Canada.

[8]York Biomedical Research Institute, Department of Biology, University of York. Wentworth Way, York, United Kingdom.

[9]Hotchkiss Brain Institute, O'Brien Institute for Public Health, University of Calgary, Calgary, Canada.

[10]These authors contributed equally: Chen Cao, Jingni He, Lauren Mak.

[#] Correspondence should be addressed to quan.long@ucalgary.ca

Abstract

DNA sequencing technologies provide unprecedented opportunities to analyze within-host evolution of microorganism populations. Often, within-host populations are analyzed via pooled sequencing of the population, which contains multiple individuals or 'haplotypes'. However, current next-generation sequencing instruments, in conjunction with single-molecule barcoded linked-reads, cannot distinguish long haplotypes directly. Computational reconstruction of haplotypes from pooled sequencing has been attempted in virology, bacterial genomics, metagenomics and human genetics, using algorithms based on either cross-host genetic sharing or within-host genomic reads. Here we describe *PoolHapX*, a flexible computational approach that integrates information from both genetic sharing and genomic sequencing. We demonstrated that PoolHapX outperforms state-of-the-art tools in the above four fields, and is robust to within-host evolution. Importantly, together with barcoded linked-reads, PoolHapX can infer whole-chromosome-scale haplotypes from 50 pools each containing 12 different haplotypes. By analyzing real data, we uncovered dynamic variations in the evolutionary processes of HIV previously unobserved in single position-based analysis.

Microorganisms are in a constant state of genetic flux in response to their environments. High-resolution analyses of these systems may lead to translational applications, such as clinical monitoring of antimicrobial resistance trends[1]. However, analyzing within-host dynamics and evolution is challenging due to the difficulty of separating samples into genetically homogeneous isolates/clones, either by experimental procedures such as culturing individual strains, or current computational tools, which are unable to distinguish between many clones when haplotypes are unknown. As a pragmatic alternative, uncultured mixtures of the heterogeneous population are sequenced and analyzed based on aggregated frequencies at each variant position in individual hosts. This procedure ignores the fact that long-range linkage information is crucial in to many different analyses towards evolution[2,3] and association mapping[4]. Recent advances in DNA sequencing technology led to increases in the sequencing depth per run and the length of reads, allowing us to assess genetic variants in greater detail, providing an unprecedented opportunity to understand the dynamics and evolution of these systems. Emergent single-molecule sequencing approaches that barcode short reads derived from long DNA fragments up to 50 Kb (i.e. linked-reads)[5-7], allow in-depth evolutionary studies into complex populations that were unresolvable from short fragment libraries (250 bp).

However, even with barcode-based linking, full resolution of short reads into long-range haplotypes is not feasible with current approaches. While third-generation technologies such as Pacific Biosciences and Oxford Nanopore Technology[8], produce very long reads that represent local haplotypes, computational analysis will still be essential for resolving haplotypes and estimating their relative proportions in the pool. Without this haplotype-level resolution, within-host dynamics cannot be analyzed as if the haplotypes were separated and sequenced individually. Even with barcoded linked-reads designed for single-molecule sequencing, the current tools are only applicable in a two-haplotype system, as it was

80 designed for analyzing paternal and maternal chromosome[9] and structural variants[10].
81 Therefore, microorganism-based studies resort to computational tools to make single-cell-like
82 analyses possible[11].
83
84 Many tools have been developed to reconstruct haplotypes using algorithms that target data
85 from viruses, bacteria, metagenomic data, and historically, artificially pooled human genomes.
86 Conceptually they can be split into two categories. The first contains statistical models utilizing
87 haplotype-block sharing between individuals ("statistical linkage disequilibrium" or "statistical
88 LD" hereafter), mostly developed in human genetics[12,13]. The second contains computational
89 algorithms that leverage minor allele frequency and sequence reads exposing the co-
90 occurrence of multiple alleles on the same haplotype (referred to as "physical linkage
91 disequilibrium" or "physical LD" hereafter), mostly tailored to uncultured sequencing of viruses
92 or bacteria[14-20]. A more detailed overview of these methods is available in the **Supplementary**
93 **Notes.**
94
95 There is a disconnect between these two approaches, and both have limitations. Methods
96 based on genetic sharing do not consider pool-specific dynamics that can be captured by
97 sequencing reads, and are ineffective when there is strong within-host evolution that changes
98 allele distributions. Also, the underlying population genetic models developed in human
99 genetics may not fit in microorganisms well. For instance, many microorganisms exchange
100 genetic materials through gene conversions[21], instead of meiotic recombination as assumed by
101 many phasing tools[22]. On the other hand, methods based on genomic reads only work on one
102 pool a time, without taking advantage of genetic sharing caused by transmission between
103 hosts[23] and common developmental trajectories in different hosts[24]. Additionally, organisms
104 with differing properties such as mutation rates generate data that require field-specific

assumptions to process. The result is that each haplotyping tool may be ineffective when applied to another data type.

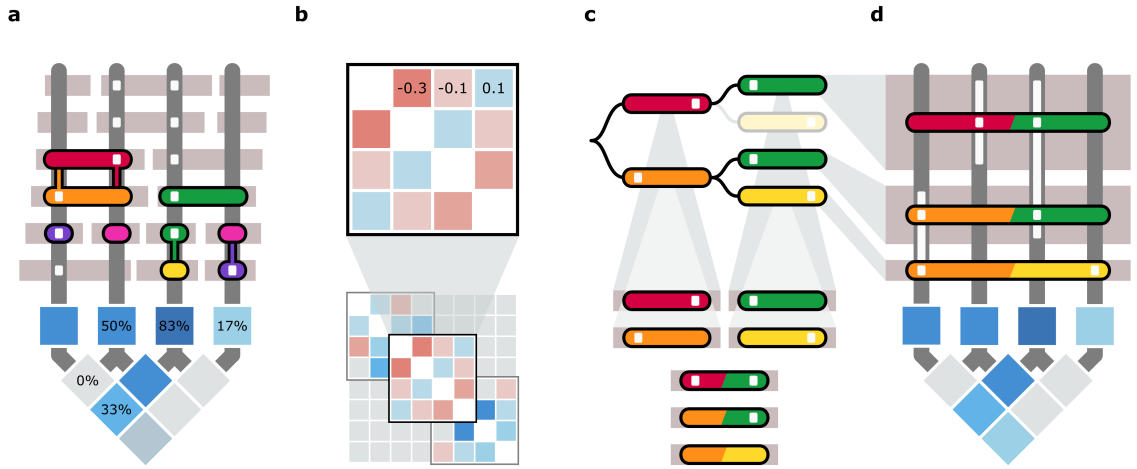
RESULTS

The two current methodologies utilize disparate sources of genetic information. We expected considerable improvements by integrating them into a reconstructive model that accurately represents the composition of genetically related populations. In this work, we present a novel tool, PoolHapX, which utilizes a multi-step framework balancing cross-host sharing and within-host specifics to best reconstruct haplotypes, regardless of the specific causes of genetic sharing (such as recombination or gene conversion) or the configuration of genomic data. Using simulated and real data containing tens of haplotypes in multiple pools, we demonstrate that PoolHapX possesses the desired properties for being a universally applicable tool robust to various factors: (1) it is applicable to many fields, and in particular outperforms state-of-the-art haplotype-reconstruction tools targeting four different domains, i.e., virus, bacteria, metagenomics and humans in their respective settings; (2) it is robust to various scenarios of within-host evolution; (3) it can reconstruct many whole-chromosome long-range haplotypes when applied to barcoded linked-reads and (4) haplotypes inferred by PoolHapX reveal novel evolutionary insights unseen in single nucleotide polymorphism (SNP)-based analyses.

Design Philosophy of PoolHapX. PoolHapX is designed to integrate cross-host sharing (i.e., statistical LD estimated from population genetic models) and within-host specifics (i.e., physical LD assessed by genomic reads) in a flexible framework. Here, “pool” and “host” are both used synonymously, since “host” emphasizes the biological source of data and “pool” the experimental source. Briefly, statistical LD in a population is equivalent to non-zero correlation

130 between the occurrences of alleles at different locations, regardless of the biological
131 mechanism (e.g., recombination, gene conversion, or natural selection). We use hierarchical
132 multivariate normal distributions to model and resolve the statistical LD using an extension of
133 the Approximate Expectation-Maximization (AEM) algorithm[25]. Physical LD can be observed in
134 paired-end reads or linked-reads, constraining the combinations of haplotypes that can occur
135 in specific pools. We utilize this information in two ways: (1) to generate initial draft haplotypes
136 and (2) to statistically infer final haplotypes using a regularized regression model.
137
138 ***Overview of the PoolHapX algorithm.*** PoolHapX infers haplotypes from genomic data that
139 contains a pool of multiple haplotypes. PoolHapX uses variant calls and read alignments to
140 identify global haplotypes in each pool, and estimate their in-pool frequencies (**Supplementary**
141 **Fig. 8**). Full mathematical descriptions of PoolHapX are presented in **Online Methods,** with
142 implementation details found in the **Supplementary Notes**. The PoolHapX algorithm is
143 comprised of four steps embedded in a divide-and-conquer framework. In step 1, graph-
144 coloring[26] is employed to roughly cluster sequencing reads into initial draft haplotypes. This
145 draft set serves as the first step of the divide-and-conquer process (**Fig. 1a, Supplementary**
146 **Fig. 1**). In step 2, our hierarchical Approximate Expectation-Maximization (AEM) algorithm is
147 applied to infer haplotypes in local regions. The algorithm starts with the smallest local
148 haplotypes as the bottom hierarchy (**Fig. 1b**), and then gradually combines them into
149 successively longer local haplotypes covering larger spans of variant positions (**Fig. 1b,**
150 **Supplementary Fig. 3**). This process iterates through several rounds until reaching the top
151 level (representing the largest local region that AEM can analyze with the available
152 computation memory). In step 3, the refined set of local haplotypes from the final iteration of
153 AEM will be stitched to form candidate global haplotypes using a Breadth-First Search (BFS)
154 algorithm (**Fig. 1c, Supplementary Fig. 6**). Finally, in step 4, the long-range linkage implied in

155 allele frequencies aggregated across all hosts and the short-range linkage sequencing reads
156 from each pool are integrated in a novel regression model (**Fig. 1d, Supplementary Fig. 7**).
157 This regression model is solved using a regularized objective function that combines L0 and L1
158 penalization[27]. This step finalizes the identity and frequency of the global haplotypes. In
159 summary, steps 1 and 4 utilize within-host specific reads, whereas step 2 iteratively leverages
160 linkage-block sharing across pools regardless of the information's biological source.
161



162

163 **Figure 1. The PoolHapX algorithm**. **(a)** An example of read-graph construction based on physical LD and allele
164 frequency calculation for single site as well as paired sites. Dark grey (vertical) bars denote polymorphic sites;
165 white squares represent non-reference alleles. Light grey (horizontal) rectangles represent reads. Each read is a
166 node in the graph. Colored columns are edges in the graph, specifying linked nodes that must be colored
167 differently. Colored round bars represent nodes that can be colored based on the information from the edges. **(b)**
168 An example of Hierarchical Approximate Expectation Maximization. The algorithm starts with a 4 x 4 variance-
169 covariance matrix, representing a local region with 4 genetic variants. It will be extended to an 8-variant region in
170 this example. The darker blue indicates higher positive statistical LD between the two genetic variants; red
171 indicates negative LD. **(c)** An example tree representing all potential global haplotypes. Local haplotypes (2 local
172 haplotypes in each region) will be stitched into three four potential global haplotypes by Breadth-First Search. **(d)**
173 An example for the regularized regression. Here four genetic variants have provided seven samples to train the
174 model: four within-host aggregated MAFs and three non-zero pair-wise physical LDs, represented by the blue
175 squares. A regularized (L0+L1) optimization will solve the haplotype frequencies, represented by the area of the
176 shaded rectangles. More detailed illustrations of the algorithms are in **Supplementary Figures 1-7**.
177

178 ***Benchmarking PoolHapX with simulated data.*** To test the accuracy and flexibility of
179 PoolHapX in comparison to state-of-the-art haplotyping tools [14-20,25,28], we simulated artificial
180 pools of haplotypes and the sequencing reads generated from these pools. We then examined
181 PoolHapX against tools developed for virology[14-16], bacteriology[17,18], metagenomics[19,20], and

182 human genetics[25,28] (flowchart in **Supplementary Fig. 8**). As each discipline has a specific
183 method for simulating data for benchmarking purposes, we follow their corresponding
184 conventions. However, for direct comparisons to PoolHapX we used standardized assessment
185 criteria (**Supplementary Notes**). We used Matthews Correlation Coefficient (MCC) to measure
186 the similarity between the identities of simulated 'gold standard' and reconstructed haplotypes,
187 where identical haplotypes have MCC = 1 (the larger, the better). We used Jensen–Shannon
188 Divergence (JSD) to measure the difference between frequencies of simulated and
189 reconstructed haplotype distributions, where identical distributions have JSD = 0 (the smaller,
190 the better). **Fig. 2** shows the results of this comparison, and a brief description of each domain
191 is provided below. Details of the simulations are presented in the **Supplementary Notes** and
192 **Online Methods**, and outcomes with more parameters, showing similar trends, are presented
193 in **Supplementary Figs. 12-14**)

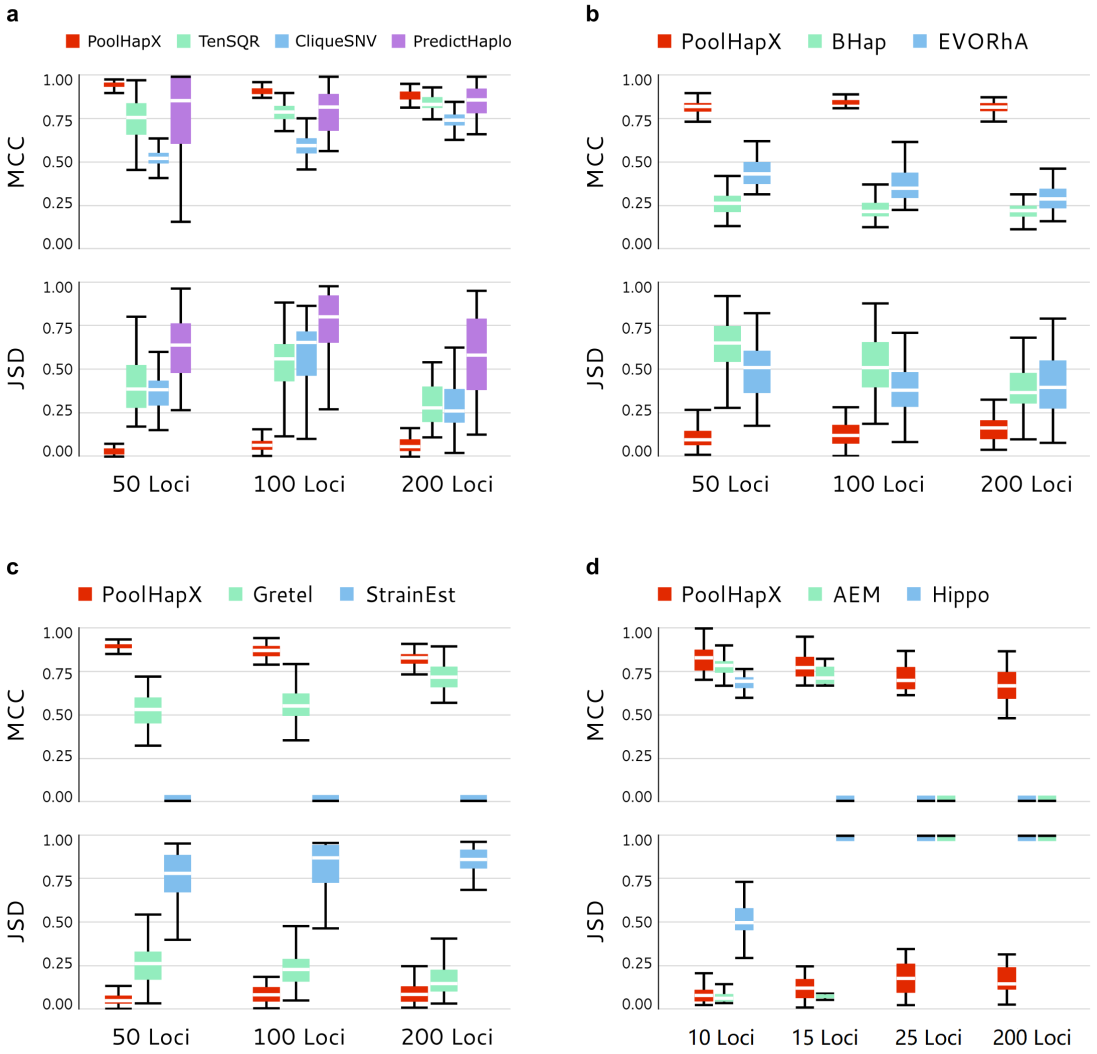


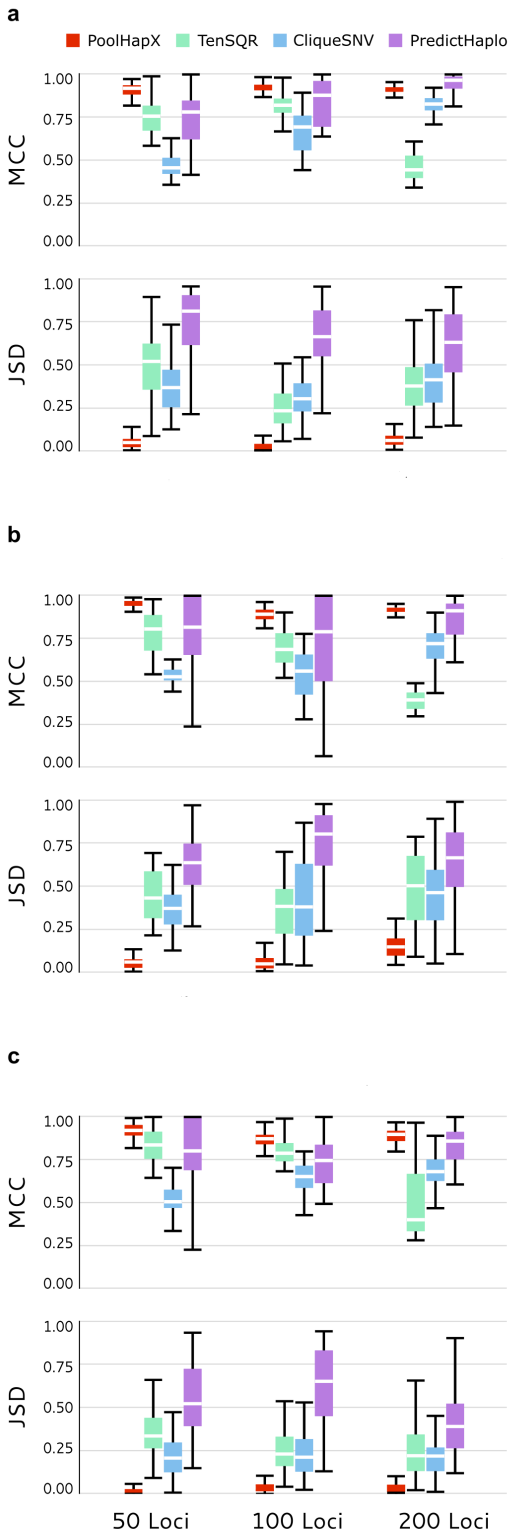
Figure 2. Comparison between PoolHapX and existing tools. For all panels, the upper half shows the accuracy for haplotype identity (MCC) and the lower half shows the accuracy for haplotype frequency (JSD). The x-axis denotes number of genetic variants in the haplotype. Boxes extend to the first and third quartile; whiskers extend to the upper and lower value. **(a-c)** Number of pools = 50. **(a)** tools to reconstruct viral haplotypes, TenSQR, PredictHaplo and CliqueSNV. Sequencing coverage per pool = 2000X. Number of cross-pool haplotypes for 50 loci, 100 loci and 200 loci are 41, 73 and 42. Average number of haplotypes per pool for 50 loci, 100 loci and 200 loci are 12, 18 and 11, respectively. **(b)** tools to reconstruct bacterial haplotypes, Bhap and EVORhA. Coverage = 250X. Number of cross-pool haplotypes for 50 loci, 100 loci and 200 loci are 42, 43 and 43. Average number of haplotypes per pool for 50 loci, 100 loci and 200 loci are 11, 11 and 12, respectively. **(c)** tools to reconstruct haplotypes for metagenomics, Gretel and StrainEst. Coverage = 100X. Number of cross-pool haplotypes for 50 loci, 100 loci and 200 loci are 39, 37 and 36. Average number of haplotypes per pool for 50 loci, 100 loci and 200 loci are 9, 10 and 10, respectively. **(d)** tools to reconstruct human haplotypes, Hippo and AEM (using 26 populations in 1000 Genome Project data). Coverage = 100X. Number of pools = 25. Number of cross-pool haplotypes for 10 loci, 15 loci, 25 loci and 200 loci are 15, 22, 28 and 35. Average number of haplotypes per pool for 10 loci, 15 loci, 25 loci and 200 loci are 4, 5, 6, and 6, respectively. The bars without whiskers represent no data.

212 *Viruses & Bacteria*. In the field of viral/bacterial haplotype reconstruction, cross-host linkage
213 sharing is not used as a source of information, despite literature evidence demonstrating
214 extensive conservation in some genomic regions even after transmission takes place[29]. As a
215 result, the authors formed only one pool of gold standard haplotypes in each round of
216 simulation. A reference genome is used as a template, and variants are randomly simulated
217 with pre-specified density of SNPs to form the gold standard haplotypes. Multiple densities are
218 used to benchmark performance with diverse configurations of data, reflecting variable
219 mutation rates in different environments[30]. In general, high SNP density (between 0.5% and
220 1.5%) for viruses [16], and a lower range for bacteria (between 0.01% and 0.05%)[18] are used.
221 Our procedure is similar, except that we simulated multiple pools with some haplotype sharing
222 between them, and reconstructed haplotypes across the pools simultaneously. We did not use
223 a commonly cited real dataset with only 5 HIV strains[31] because PoolHapX is designed to
224 utilize multiple pools with genetic sharing and many haplotypes in each pool.
225
226 Due to our approach, increased haplotype sharing between pools led to better performance for
227 PoolHapX, but not for other tools. To accurately simulate linkage sharing between pools, we
228 used SLiM[32] to simulate haplotypes under standard island models, where genomes mutate,
229 recombine, and replicate in their own island and occasionally migrate to other islands. We
230 embedded the simulated variants into a viral reference genome.
231
232 For viruses, we chose the human immunodeficiency virus (HIV), which is well known for its
233 ability to form large and genetically heterogeneous within-patient viral populations[33]. Multiple
234 haplotypes were pooled and the simulated sequencing reads were processed using a modified
235 version of the GATK best practice pipeline[34] to discover variants. Details can be found in
236 **Supplementary Notes**. We chose three representative viral sequencing tools for comparison:

237 TenSQR[14], PredictHaplo[16] and CliqueSNV[15]. Evidently, PoolHapX outperformed these
238 alternatives not only in the mean of the MCC and JSD values, but also their variances (**Fig. 2a**
239 and **Supplementary Fig. 12**). When sequencing coverage was high (=1000X), as well as the
240 SNP density, PoolHapX performed similarly to other tools in terms of MCC, but with much
241 better JSD (**Fig. 2a**). When sequencing coverage was low, the performance of other tools
242 decreased rapidly relative to PoolHapX (**Supplementary Fig. 12**).
243
244 For bacteria, we used a chromosome of *Vibrio cholerae* O1 biovar *El Tor* str. N16961 (Chr-2,
245 length = 1.07Mb), a strain of the bacterium *Vibrio cholerae* and the causative agent of
246 cholera[35] as the template reference genome. We used a lower SNP density (=0.005% to
247 0.02%) to match the bacterial genomes[17] and to be comparable to the simulation procedures
248 of competing tools, i.e., Bhap[18] and EVORhA[17]. Despite the very different simulation
249 parameters and reference genome in the simulations of viruses and bacteria, we observed
250 similarly good performance in contrast to other tools (**Fig. 2b** and **Supplementary Fig. 12**).
251
252 *Metagenomics*. In the field of metagenomics there is no established tool to infer haplotypes *de*
253 *novo* without utilizing template references of known strain sequences. Gretel, a recently
254 developed tool[20], requires high SNP density so that each SNP is within a sequencing read
255 length of the next adjacent SNP. We simulated data to facilitate the requirements of Gretel. We
256 also selected StrainEst[19], a representative tool for strain-identification, to demonstrate whether
257 tools that utilize templates of known strains may work for fine-scale haplotype identification.
258 This was likely an unfair comparison, due to the lack of template references for known strains
259 in our simulations. We used sequences from *Escherichia coli*, a typical bacterium for meta-
260 genomics studies. In these two cases, the sequencing coverage was substantially lower than
261 dedicated single-species sequencing (=50X). Evidently, PoolHapX outperformed Gretel, and

262 StrainEst did not work well when templates were unavailable (**Fig. 2c; Supplementary Fig.**
263 **12**).
264
265 *Humans*. In early genome-wide association studies (GWAS) for humans, DNA from multiple
266 individuals was artificially pooled to save on genotyping costs. Subsequently, *in silico* methods
267 were applied to the pooled sequencing data to reconstruct haplotypes[25,28] for association
268 mapping. Though technological advancements have made this cost-saving practice
269 unnecessary, as a theoretical assessment, we compared PoolHapX against the GWAS-based
270 haplotype reconstruction tools Hippo[28] and AEM[25]. Since there are many publicly available
271 human genomes we did not simulate haplotypes, instead making artificial pools using phased
272 haplotypes from the 1000 Genomes Project[36] (**Supplementary Notes**). **Fig. 2d** and
273 **Supplementary Fig. 14** show that PoolHapX slightly outperformed alternative tools when
274 there were relatively few SNPs. When there were many SNPs (>=25) in a region, however, the
275 other tools did not finish in two weeks, while PoolHapX could still produce reliable results with
276 a large region containing as many as 200 SNPs in a few hours.
277
278 ***PoolHapX performance in evolutionary scenarios and linked reads.*** The above
279 simulations in four domains show that PoolHapX is universally applicable across species and
280 is robust to sequencing data with varying properties. However, except in the domain of human
281 GWAS, existing tools do not explicitly utilize sharing across hosts. Since different hosts of
282 microbes may have different evolutionary patterns, leading to host-specific haplotypes, this can
283 cause a biased outcome from models that use cross-host sharing. To test whether the
284 PoolHapX module utilizing within-host physical LD can correct for this bias, we used SLiM to
285 simulate data under three common evolutionary scenarios in population genetic analyses:
286 positive selection, negative selection, and selective sweeps (**Supplementary Notes**). Our

analysis of these data indicated that PoolHapX is robust to host-specific haplotype patterns caused by evolution (**Fig. 3; Supplementary Fig. 13**).



291 **Figure 3. PoolHapX is robust to the within-host changes due to selective forces.** The three evolutionary
292 forces are: **(a)** Negative selection, number of cross-pool haplotypes for 50 loci, 100 loci and 200 loci are 36, 36
293 and 45. Average number of haplotypes per pool for 50 loci, 100 loci and 200 loci are 11, 11 and 12, respectively,
294 **(b)** Positive selection, number of cross-pool haplotypes for 50 loci, 100 loci and 200 loci are 41, 50 and 104.
295 Average number of haplotypes per pool for 50 loci, 100 loci and 200 loci are 12, 14 and 21, respectively, and **(c)**
296 Selective Sweep, number of cross-pool haplotypes for 50 loci, 100 loci and 200 loci are 23, 36 and 27. Average
297 number of haplotypes per pool for 50 loci, 100 loci and 200 loci are 8, 9 and 9, respectively. All three panels are
298 comparing with viral tools, i.e., TenSQR, PredictHaplo and CliqueSNV. All data are simulated under coverage of
299 1000X, and 50 pools. The y-axis is the same as **Fig. 2**.
300

301 *Single-molecule linked-reads*. We further tested PoolHapX's capabilities on single- *molecule*
302 linked-reads. Based on a template of chromosome 1 of the unicellular green algae
303 *Ostreococcus lucimarinus* (genome length of 1.15 Mb), we simulated approximately 20 gold
304 standard haplotypes with 570 SNP positions. Using the 10X Genomics linked-read simulator
305 LRSim[37], with default settings of fragment length (=50Kb) in each droplet and number of
306 linked-reads per fragment (on average 67), we simulated 10X Genomics linked-reads at
307 various sequencing depths and numbers of pools. On average, PoolHapX achieved MCC ≥
308 0.75 and JSD ≤ 0.25 (**Fig. 4**), which is comparable to other PoolHapX results when inferring
309 shorter haplotypes using standard Illumina paired-end reads based on short-fragment DNA
310 molecules. This outcome turns the promise of "single-cell" DNA sequencing into reality,
311 enabling pathogen biologists to study within-host evolutionary changes at the cellular level.
312

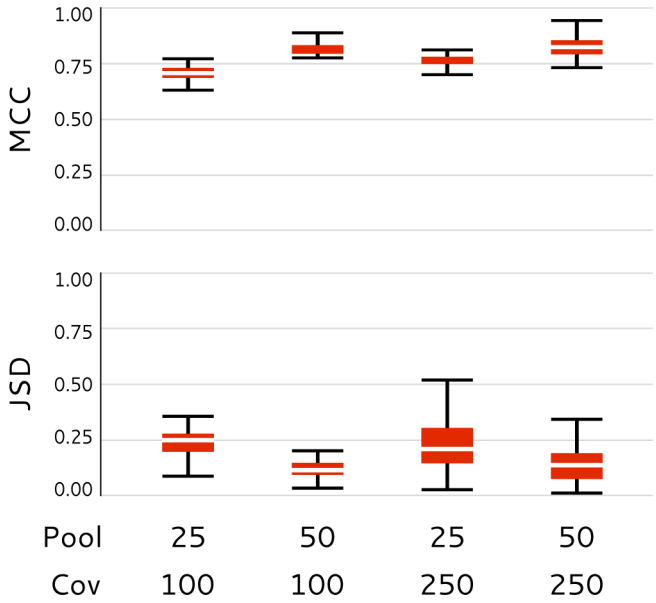


Figure 4. PoolHapX + single-molecule linked-reads. MCC and JSD of PoolHapX applying to simulated barcoded linked-reads based on combinations of different numbers of pools (25,50) and sequencing coverage (100, 250). Number of cross-pool haplotypes for 25 and 50 pools are 31 and 40. Average haplotypes per pool for 25 and 50 pools are 12.

Applications to real data. Infections with the haploid malaria parasite *Plasmodium vixax* (Genome size =22 Mb) are known to contain multiple genotypes, which influence disease severity[38]. These 'multi-clonal infections' may derive from infection by a single mosquito bite carrying multiple strains, with meiotic recombination in the vector. Alternatively, they may be due to multiple infections from different mosquitos carrying different strains. Accurate whole-genome haplotype reconstruction will distinguish between these alternatives. We challenged PoolHapX with a collection of 49 *P. vivax* genome sequences (**Supplementary Notes**) to demonstrate its applicability on many pools of eukaryotic organisms with mid-sized genomes[39]. To achieve this, we split the *P. vivax* genome into windows of 150 SNPs. PoolHapX took on average 54.79 CPU-hours per chromosome to conduct all computations. On average we found 3.3 inferred haplotypes per region per individual (**Supplementary Table 3**), consistent to

331 expectations[38]. The distribution of haplotype frequencies along the chromosomes is
332 exemplified in **Supplementary Fig. 9.**
333
334 To compare the *de novo* reconstructed haplotypes with strains inferred by a template-based
335 method in metagenomics, we reanalyzed a meta-genomics dataset collected from a
336 gastrointestinal microbiome undergoing shifts in species and strain abundance[40]. The original
337 publication suggested that the abundance of *Staphylocous epidermidis* is primarily controlled
338 by phages 13, 14 and 46 through the *mecA* gene. Based on StrainEst[19] (supported by
339 templates of known strains), other researchers analyzed the same data and inferred the
340 identities and frequencies of the three strains in question[19], which we were able to replicate
341 (**Fig. 5.a**). We have re-analyzed the same data by reconstructing fine-scale haplotypes.
342 The *Staphylocous epidermidis* genome was divided into 110 fragments (100 SNPs
343 per fragment). The average number of haplotypes for each fragment was 9.3, although this
344 value changed at different time points (**Supplementary Table 4**). All fragmentary haplotypes
345 are aligned back to the three main strains (**Supplementary Notes**) to examine the aggregated
346 haplotype frequencies of each strain. By averaging all 110 regions, the aggregated frequencies
347 (from PoolHapX) were found to follow the same pattern of changes as these three strains (**Fig.**
348 **5.b**). This demonstrates that PoolHapX correctly identified haplotypes through *de novo*
349 inference, without the use of reference templates from known strains, as required by StrainEst.
350

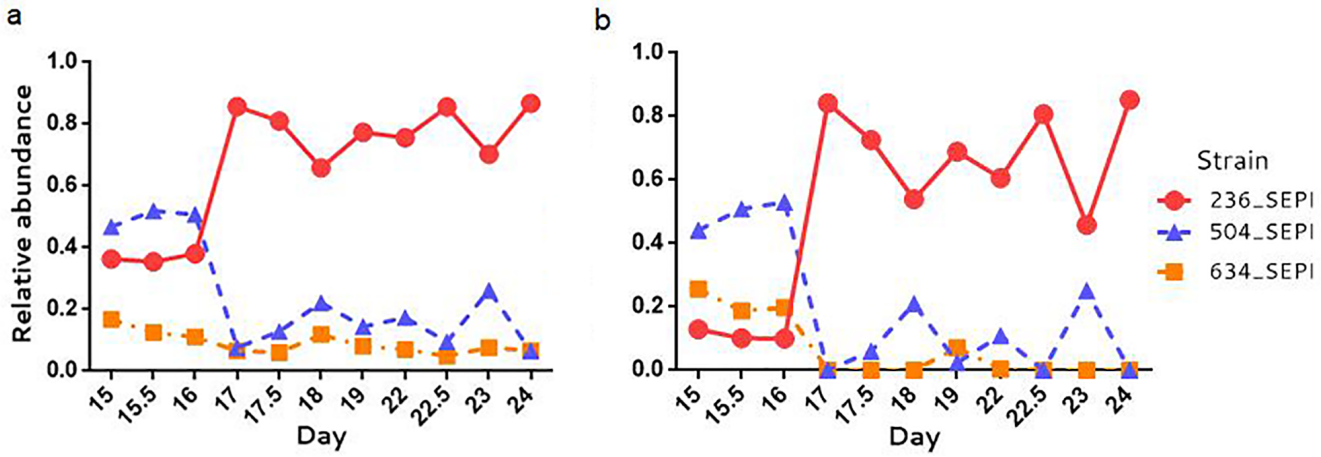
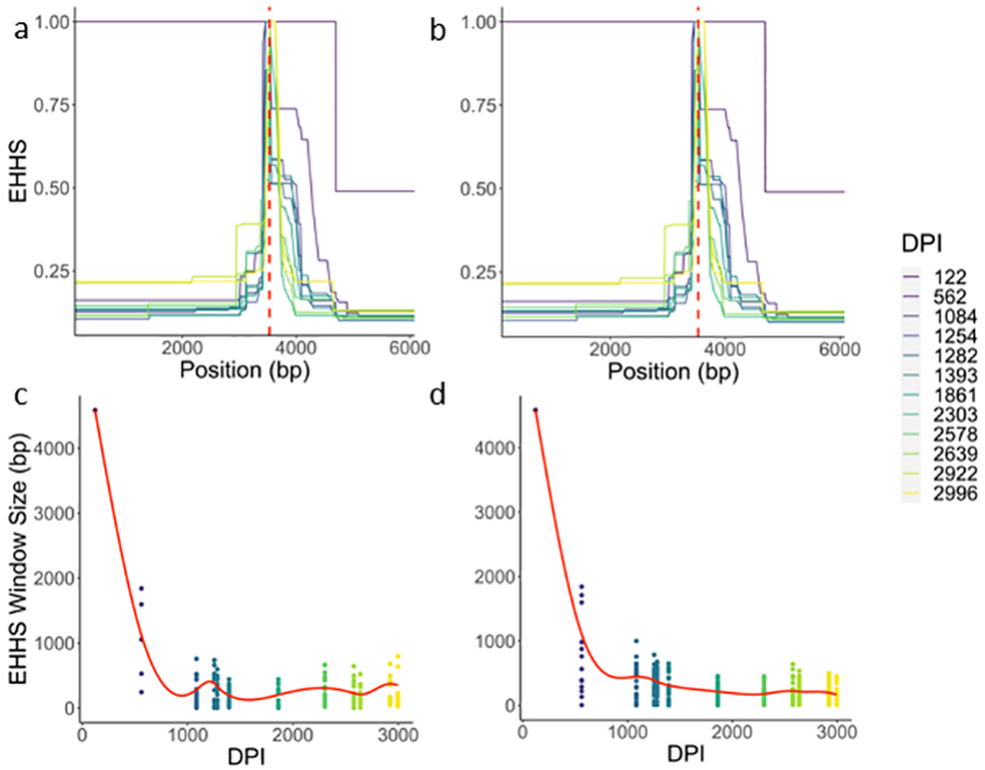


Figure 5. ***Staphylococcus epidermidis*** **strain abundance** calculated *de novo* by PoolHapX **(a)** and StrainEst based on templates of known strain **(b)** for early stages of infant gut colonization. All haplotypes predicted by PoolHapX are aligned to the three strains and we observe the same pattern of the changes of these three strains.

To demonstrate how PoolHapX can be used to discover novel evolutionary events, we tested PoolHapX on bulk-sequencing data from a recent intra-patient HIV study[41]. This dataset contains longitudinal samples from multiple time points for 10 patients. We analyzed patient #1, which contains the most time points (12). We inferred haplotypes using PoolHapX, and observed 2 main haplotypes at time point 1, and 10 -13 main haplotypes at the rest time points (**Supplementary Table 5**). We then calculated several extended haplotype homozygosity-related (EHH) summary statistics[2], which measure linkage disequilibrium across a population by quantifying the probability that two randomly chosen particles are identical by descent in a certain region (see rationale in **Supplementary Notes**). Outlier values of the area under the EHH curve indicate that selective sweeps may have occurred (**Supplementary Notes**). While the size of linkage blocks decayed extremely rapidly post-infection in all genes (**Fig. 6a,b, Supplementary Fig. 10**), it did not decrease monotonically as the HIV population adapted to the within-patient environment. To further quantify the rate and dynamics of selection within each gene, we plotted the size of windows with EHHS ≥ 0.5 at all time points and for multiple genes in the reconstructed haplotypes. The genes *gag*, responsible for assembly and structure, and *pol*, responsible for genetic reproduction[42], are pictured in **Fig. 6c, d**, (other

372 genes in **Supplementary Fig. 11**). Within *gag* and *pol*, there was substantial heterogeneity in
373 average window size over time, with the downstream regions of *gag* and *pol* largely fluctuating
374 between 0 and 250 bp (**Fig. 6c,d**). These regions were highly conserved due to their
375 respective roles in the HIV life cycle[43].
376



377
378 **Figure 6. Insights in within-host HIV evolution (a-b)** The decay of EHHS around each SNP position in
379 reconstructed HIV-1 haplotypes occurs rapidly during the acute phase of infection. The dashed red line indicates
380 the location of the focal SNP position. **(a)** Position 1377 (Gag gene, found in p2 protein). **(b)** Position 3530 (Pol
381 gene, found in p15). **(c-d)** The size of windows of EHHS 0.5 fluctuate around gene-specific averages. The solid
382 red line indicates a weighted mean across the positions in the gene. DPI refers to estimated days post-infection.
383 Each dot represents the window size around at least one position. **(c)** Gag. **(d)** Pol. The legend (right) indicating
384 the color corresponding to each time-point is common to panels **a-d**.
385

386 We have conducted a search for regions of positive selection between reconstructed
387 haplotypes at adjacent time-points, where selective sweeps could have taken place. There are
388 regions that are recurrently swept, most notably in the region of the *gag* polyprotein gene that
389 encodes the p24 protein (**Supplementary Table 7**). The occurrence of sporadic but re-
390 occurring selective sweeps in gag, specifically p24, can be attributed to the appearance of

391 cytotoxic T-lymphocytes (CTL) escape mutations, which reduce the ability of CTL to target
392 virus-infected cells[44]. However, these escape mutations also decrease the replicative capacity
393 of the virus, and a larger mutational burden corresponds to a greater decrease in capacity[45,46].
394 As such, episodic periods of positive selection at the same location would allow successful
395 escape mutations to rise to fixation occasionally, while still allowing for genetic diversity to
396 accumulate between selective sweeps.
397
398 **DISCUSSION**
399
400 We have shown that PoolHapX produces more accurate haplotype sequences and
401 frequencies than any other tool available to date, and is robust to dynamics caused by within-
402 host evolution. From the analysis of *Plasmodium vivax, Staphylocous epidermidis* and HIV
403 data, we show that PoolHapX is scalable, accurate, and infers haplotype data that is valuable
404 for understanding the within-patient diversity of pathogens.
405
406 This implementation of PoolHapX has some limitations. We found that the method is sensitive
407 to the inferred allele frequency, and therefore high variance in allele frequency caused by very
408 low sequencing coverage will result in high error rates. The performance of PoolHapX is also
409 variable when we attempt to infer frequencies of more than 50 haplotypes in the pools.
410 However, if we aim only to assess a smaller number of more abundant haplotypes (e.g. 10-
411 20), it is robust to noise caused by rare haplotypes (**Fig. 2** and **3**). Several existing papers use
412 L1 regularization alone[19]. This strategy does not work for many haplotypes with small
413 differences. This is because the sum of haplotype frequencies in a host will always be near 1.0
414 if the inference is generally accurate. L0 regularization controls for this problem by further
415 regularizing the number of haplotypes. At present, PoolHapX is in continuing development,

416 with ongoing work to integrate third-generation sequencing data[47] into PoolHapX, as well as
417 algorithms using genomic assembly[48] to improve haplotypes.
418
419 **METHODS**
420
421 **Graph coloring algorithm**. If two sequencing reads cover the same genetic variant site but
422 carry different alleles, it is certain that they do not belong to the same haplotype. Based on this
423 observation, we build a graph <V, E> in which every read is a node v. For two nodes v_1 and v_2,
424 we put an edge $e_{1,2}$ between them if and only if we have information to claim they are not on
425 the same haplotype. Then, the haplotyping problem becomes a graph-coloring problem: where
426 each node (i.e. each read) is assigned a color, such that nodes linked by edges are colored
427 differently. This ensures that reads belonging to different haplotypes are colored differently.
428 After conducting this graph-coloring problem, we effectively estimate haplotypes by collecting
429 reads of the same colors. As the standard parsimony algorithm is too slow when the number of
430 reads is large, we have implemented a greedy algorithm to color this graph (**Supplementary**
431 **Notes**). The spatial complexity of our graph coloring is $O(n^2)$ and the time complexity is $O(n^2)$.
432 The outcome of graph-coloring forms starting states for the whole pipeline in two respects.
433 First, by collecting all reads of the same color, PoolHapX can generate segments of local
434 haplotypes as the initial input to the next step, the Hierarchical AEM algorithm. Second, the
435 gaps between local haplotypes naturally inform the initial divide & conquer plan for subsequent
436 steps (**Supplementary Notes**).
437
438 **Hierarchical AEM algorithm**. The basic version of AEM algorithm, as described in[25], builds
439 upon the multivariate normal (MVN) distribution. The LD between all pairs of *n* genetic variants
440 is modeled as the variance-covariance matrix of an MVN distribution. Initially, all 2^n possible

441 haplotypes will be assigned to the same frequency ($1/2^n$). Then in an iterative procedure, the
442 likelihood ratio of observing the data with or without the presence of each haplotype is
443 estimated using the MVN densities.. These ratios are called "Importance factors"[25], indicating
444 the importance of the individual haplotypes, and will be used to adjust their haplotype
445 frequencies. This adjustment is conducted iteratively until the haplotype frequencies converge
446 (**Supplementary Notes**).
447
448 In our adaptation of AEM we have made the following three modifications. First, the initial
449 haplotype configuration is no longer a uniform distribution of all 2^n haplotypes. Instead, using
450 the haplotypes gained from graph coloring, haplotypes with higher sequence coverage start
451 with a higher initial frequency. As a consequence, many potential haplotypes that are not
452 observed in graph coloring will have zero frequency. While the spatial complexity of AEM
453 remains $O(n^2)$ and the theoretical time complexity remains $O(n^3 \times 2^n)$, the number of required
454 iterations are substantially reduced in practice due to the initial configuration being closer to
455 the truth. Second, we use a divide & conquer algorithm to scale up the original AEM algorithm
456 to larger regions, so that we run AEM in a hierarchical manner. The shorter haplotypes inferred
457 from the previous AEM iteration are used in the next round of AEM to form longer haplotypes
458 (**Supplementary Fig. 3 and Fig. 1a**). In each round, local regions are designed to have half of
459 their genetic variants overlap with the next region (**Supplementary Fig. 5**), in order to form
460 tiling windows that can be stitched together at the next hierarchical level. Third, the original
461 AEM is not robust to numerical instability if the denominator in the likelihood ratio is close to
462 zero. However, this problem occurs more frequently in larger regions with sparse non-zero LDs
463 in the covariance matrix. We have fixed this by adjusting the calculation of the likelihood
464 (**Supplementary Notes**).
465

466 **Breadth-First Search (BFS)**. The iterative AEM algorithm generates successively larger
467 regional haplotypes until an upper limit is reached, which is 96 variants by default. PoolHapX
468 will then attempt to resolve global haplotypes. The outcome of the last AEM iteration is a set of
469 local haplotypes that span tiling windows, with many potential combinations that form global
470 haplotypes. To resolve global haplotypes, we model the local regions as a tree, with each local
471 haplotype as a node. Haplotypes from the first region of the genome form the first level of the
472 tree, while haplotypes from the next tiling region form the nodes of the next level. If two
473 haplotypes in adjacent regions have the same alleles in their overlapping segments, we add an
474 edge linking these two nodes. Traversing the resulting tree generates an exhaustive set of all
475 plausible combinations of local haplotypes, which forms the set of candidate global haplotypes.
476 We implement a standard Breadth-First Search (BFS) algorithm[49] to conduct this traversal.
477 Finally, we filter out some global haplotypes that are inconsistent with the sequencing reads
478 (**Supplementary Notes**).
479
480 **Global Regularization model**. Given all candidate global haplotypes from the BFS step, we
481 use an innovative regression model to estimate the within-host global haplotype frequencies in
482 each pool:
483

$$Y \sim \sum \beta_i X_i$$

484
485 Where β_i is the frequency of the *i*-th global haplotype in the host (pool). Here Y and X_i are the
486 independent and predictor variables in a standard regression model. We use two types of
487 samples to train Y from X_i, which represent two different sources of data: minor allele
488 frequency and physical LD. Mathematically, the dimension of Y is *n* + *n(n-1)/2* (where *n* is the

number of sites), representing the alternate frequency at *n* sites and their *n(n-1)/2* physical LD across pairs of sites observed in the reads. First, at each site, the sum of frequencies of haplotypes containing the alternate allele should be equal to the observed alternate allele frequency based on reads from the pool. This is the same information utilized by several other tools[17,19]. An innovation of our design is the use of a second type of sample: for each pair of sites, the sum of frequencies of haplotypes containing both alternate alleles should be equal to the frequency observed in the number of reads that cover both alternate alleles in the pool, which includes read-pairs and many barcoded reads in 10x linked-reads (**Supplementary Fig. 7**). A full description is in (**Supplementary Notes**). The set of β_i that best fits these two sets of constraints is our solution.

To reduce overfitting, the objective function for training the above regression is designed as a combination of L0 and L1:

$$Obj(\vec{\boldsymbol{\beta}}) = \left\|Y - \hat{Y}\right\|_2 + \alpha||\vec{\boldsymbol{\beta}}||_0 + \gamma||\vec{\boldsymbol{\beta}}||_1$$

where $||\vec{\boldsymbol{\beta}}||_1$ is the L1-norm, which is the sum of absolute values of all β_i; and $||\vec{\boldsymbol{\beta}}||_0$ is the L0-norm, which is the number of non-zero β_i.

As mentioned in the **Discussion**, L1 regularization cannot differentiate between different haplotypes sets that are distinguished by small differences such as a single recombination. This is why L0 regularization is necessary for our method, although it is much slower. The

511 L0Learn package[27] is used to conduct this inference. Finally, the cross-host (i.e., population)
512 frequencies of each haplotype can be formed by combining the within-host frequencies.
513
514
515 **HIV evolutionary data analysis**
516
517 For a description of Patient 1 data, the SNP position-calling pipeline, and haplotype
518 reconstruction, see **Supplementary Notes**.
519
520 The R package rehh (version 3.0) was applied to survey linkage patterns within a single time-
521 point, and changes to linkage patterns across the duration of infection monitoring[50]. Several
522 long-range haplotype-based evolutionary statistics related to extended haplotype
523 homozygosity (EHH)[2] were used to quantify the type and magnitude of selection. To search for
524 regions of positive selection within the reconstructed genome, integrated haplotype score
525 (iHS)[3] and cross-population EHH scores (XP-EHH) were calculated for each time-point and
526 between each time-point, respectively.
527
528 For more details about the rationale behind each layer of analysis, see **Supplementary Notes**.
529 The scripts that generate ms[51] output format from PoolHapX output files and apply EHH-based
530 statistics to the reconstructed haplotypes are available at
531 (https://github.com/theLongLab/PoolHapX/tree/master/Simulation_And_Analysis/HIV_analysis_
532 code). Parameters and settings are described in further detail within the scripts.
533

Other data analyses. Processing and analyses of *Plasmodium* and other metagenomic data (*E. Coli*) can be found in **Supplementary Notes**. Details of simulations and comparisons (including how other tools are executed) are also included in **Supplementary Notes**.

Acknowledgement. Q.L. is supported by an NSERC Discovery Grant (RGPIN-2017-04860), a Canada Foundation for Innovation JELF grant (36605), and an ACHRI Startup grant. C.C., M.L. and L.M. are supported by ACHRI scholarship. L.M. is supported by a QEII award. G.Y. is supported by an NSERC Discovery Grant (RGPIN/04246-2018).

Author contributions. Conceived the project: QL and DJ; Designed the algorithms: QL, CC, and JW; Implemented the software: CC, QL and JW; Tested the software: LM, ML, DP and SG. Simulated data analyses: CC and JH; Real data analyses: LM, CC, and DP; Contributed data and advice: DJ, TM, SM, LS, MG, GY; Fine-art figures: DK; Write the manuscript: QL, CC, LM, JH, and DK, with contributions from all co-authors.

550 **Reference**

551 1. Hofer, U. The cost of antimicrobial resistance. *Nat Rev Microbiol* **17**, 3 (2019).
552 2. Sabeti, P.C. et al. Detecting recent positive selection in the human genome from haplotype
553 structure. *Nature* **419**, 832-7 (2002).
554 3. Voight, B.F., Kudaravalli, S., Wen, X. & Pritchard, J.K. A map of recent positive selection in
555 the human genome. *PLoS Biol* **4**, e72 (2006).
556 4. Datta, A.S. & Biswas, S. Comparison of haplotype-based statistical tests for disease
557 association with rare and common variants. *Brief Bioinform* **17**, 657-71 (2016).
558 5. Zheng, G.X. et al. Haplotyping germline and cancer genomes with high-throughput linked-
559 read sequencing. *Nat Biotechnol* **34**, 303-11 (2016).
560 6. Chen, Z. et al. Ultra-low input single tube linked-read library method enables short-read
561 NGS systems to generate highly accurate and economical long-range sequencing
562 information for de novo genome assembly and haplotype phasing. *bioRxiv*, 852947 (2019).
563 7. Wang, O. et al. Efficient and unique cobarcoding of second-generation sequencing reads
564 from long DNA molecules enabling cost-effective and accurate sequencing, haplotyping,
565 and de novo assembly. *Genome Res* **29**, 798-808 (2019).
566 8. Weirather, J.L. et al. Comprehensive comparison of Pacific Biosciences and Oxford
567 Nanopore Technologies and their applications to transcriptome analysis. *F1000Res* **6**, 100
568 (2017).
569 9. Mostovoy, Y. et al. A hybrid approach for de novo human genome sequence assembly and
570 phasing. *Nat Methods* **13**, 587-90 (2016).
571 10. Elyanow, R., Wu, H.T. & Raphael, B.J. Identifying structural variants using linked-read
572 sequencing data. *Bioinformatics* **34**, 353-360 (2018).
573 11. Danko, D.C., Meleshko, D., Bezdan, D., Mason, C. & Hajirasouliha, I. Minerva: an alignment-
574 and reference-free approach to deconvolve Linked-Reads for metagenomics. *Genome Res*
575 **29**, 116-124 (2019).
576 12. Long, Q. et al. PoolHap: inferring haplotype frequencies from pooled samples by next
577 generation sequencing. *PLoS One* **6**, e15292 (2011).
578 13. Long, Q. Computational Haplotype Inference from Pooled Samples. *Methods Mol Biol* **1551**,
579 309-319 (2017).
580 14. Ahn, S., Ke, Z. & Vikalo, H. Viral quasispecies reconstruction via tensor factorization with
581 successive read removal. *Bioinformatics* **34**, i23-i31 (2018).
582 15. Knyazev, S. et al. CliqueSNV: Scalable Reconstruction of Intra-Host Viral Populations from
583 NGS Reads. *bioRxiv*, 264242 (2018).
584 16. Prabhakaran, S., Rey, M., Zagordi, O., Beerenwinkel, N. & Roth, V. HIV haplotype inference
585 using a propagating dirichlet process mixture model. *IEEE/ACM transactions on*
586 *computational biology and bioinformatics* **11**, 182-191 (2013).
587 17. Pulido-Tamayo, S. et al. Frequency-based haplotype reconstruction from deep sequencing
588 data of bacterial populations. *Nucleic Acids Res* **43**, e105 (2015).
589 18. Li, X., Saadat, S., Hu, H. & Li, X. BHap: a novel approach for bacterial haplotype
590 reconstruction. *Bioinformatics* **35**, 4624-4631 (2019).
591 19. Albanese, D. & Donati, C. Strain profiling and epidemiology of bacterial species from
592 metagenomic sequencing. *Nat Commun* **8**, 2260 (2017).
593 20. Nicholls, S.M. et al. Recovery of gene haplotypes from a metagenome. *BioRxiv*, 223404
594 (2019).
595 21. Santoyo, G. & Romero, D. Gene conversion and concerted evolution in bacterial genomes.
596 *FEMS Microbiol Rev* **29**, 169-83 (2005).

597 22. Browning, S.R. & Browning, B.L. Haplotype phasing: existing methods and new
598 developments. *Nat Rev Genet* **12**, 703-14 (2011).
599 23. Cudini, J. et al. Human cytomegalovirus haplotype reconstruction reveals high diversity due
600 to superinfection and evidence of within-host recombination. *Proc Natl Acad Sci U S A* **116**,
601 5693-5698 (2019).
602 24. Toprak, E. et al. Evolutionary paths to antibiotic resistance under dynamically sustained
603 drug selection. *Nat Genet* **44**, 101-5 (2011).
604 25. Kuk, A.Y., Zhang, H. & Yang, Y. Computationally feasible estimation of haplotype frequencies
605 from pooled DNA with and without Hardy-Weinberg equilibrium. *Bioinformatics* **25**, 379-
606 86 (2009).
607 26. Matula, D.W., Marble, G. & Isaacson, J.D. Graph coloring algorithms. in *Graph theory and*
608 *computing* 109-122 (Elsevier, 1972).
609 27. Hazimeh, H. & Mazumder, R. Fast best subset selection: Coordinate descent and local
610 combinatorial optimization algorithms. *arXiv preprint arXiv:1803.01454* (2018).
611 28. Pirinen, M. Estimating population haplotype frequencies from pooled SNP data using
612 incomplete database information. *Bioinformatics* **25**, 3296-302 (2009).
613 29. Mak, L. et al. Evaluation of A Phylogenetic Pipeline to Examine Transmission Networks in A
614 Canadian HIV Cohort. *Microorganisms* **8**(2020).
615 30. Metzgar, D. & Wills, C. Evidence for the adaptive evolution of mutation rates. *Cell* **101**, 581-
616 4 (2000).
617 31. Giallonardo, F.D. et al. Full-length haplotype reconstruction to infer the structure of
618 heterogeneous virus populations. *Nucleic Acids Res* **42**, e115 (2014).
619 32. Haller, B.C. & Messer, P.W. SLiM 3: Forward Genetic Simulations Beyond the Wright-Fisher
620 Model. *Mol Biol Evol* **36**, 632-637 (2019).
621 33. Lauring, A.S. & Andino, R. Quasispecies theory and the behavior of RNA viruses. *PLoS*
622 *Pathog* **6**, e1001005 (2010).
623 34. DePristo, M.A. et al. A framework for variation discovery and genotyping using next-
624 generation DNA sequencing data. *Nat Genet* **43**, 491-8 (2011).
625 35. Cvjetanovic, B. & Barua, D. The seventh pandemic of cholera. *Nature* **239**, 137-138 (1972).
626 36. Consortium, G.P. A global reference for human genetic variation. *Nature* **526**, 68-74 (2015).
627 37. Luo, R., Sedlazeck, F.J., Darby, C.A., Kelly, S.M. & Schatz, M.C. LRSim: A Linked-Reads
628 Simulator Generating Insights for Better Genome Partitioning. *Comput Struct Biotechnol J*
629 **15**, 478-484 (2017).
630 38. Pacheco, M.A. et al. Multiplicity of Infection and Disease Severity in Plasmodium vivax. *PLoS*
631 *Negl Trop Dis* **10**, e0004355 (2016).
632 39. Carlton, J. The Plasmodium vivax genome sequencing project. *Trends Parasitol* **19**, 227-31
633 (2003).
634 40. Sharon, I. et al. Time series community genomics analysis reveals rapid shifts in bacterial
635 species, strains, and phage during infant gut colonization. *Genome Res* **23**, 111-20 (2013).
636 41. Zanini, F. et al. Population genomics of intrapatient HIV-1 evolution. *Elife* **4**(2015).
637 42. Konnyu, B. et al. Gag-Pol processing during HIV-1 virion maturation: a systems biology
638 approach. *PLoS Comput Biol* **9**, e1003103 (2013).
639 43. Mayrose, I. et al. Synonymous site conservation in the HIV-1 genome. *BMC Evol Biol* **13**, 164
640 (2013).
641 44. Prince, J.D., Walkup, J., Akincigil, A., Amin, S. & Crystal, S. Serious mental illness and risk of
642 new HIV/AIDS diagnoses: an analysis of Medicaid beneficiaries in eight states. *Psychiatr*
643 *Serv* **63**, 1032-8 (2012).

644 45. Chopera, D.R. et al. Transmission of HIV-1 CTL escape variants provides HLA-mismatched
645 recipients with a survival advantage. *PLoS Pathog* **4**, e1000033 (2008).
646 46. Wright, J.K. et al. Impact of HLA-B*81-associated mutations in HIV-1 Gag on viral
647 replication capacity. *J Virol* **86**, 3193-9 (2012).
648 47. Check Hayden, E. Genome sequencing: the third generation. *Nature* **457**, 768-9 (2009).
649 48. Bankevich, A. et al. SPAdes: a new genome assembly algorithm and its applications to
650 single-cell sequencing. *J Comput Biol* **19**, 455-77 (2012).
651 49. Cormen, T.H., Leiserson, C.E., Rivest, R.L. & Stein, C. *Introduction to algorithms*, (MIT press,
652 2009).
653 50. Gautier, M. & Vitalis, R. rehh: an R package to detect footprints of selection in genome-wide
654 SNP data from haplotype structure. *Bioinformatics* **28**, 1176-1177 (2012).
655 51. Ewing, G. & Hermisson, J. MSMS: a coalescent simulation program including recombination,
656 demographic structure and selection at a single locus. *Bioinformatics* **26**, 2064-5 (2010).
657