`HaploCart`: Human mtDNA Haplogroup Classification Using a Pangenomic Reference Graph.

Joshua Daniel Rubin[1*], Nicola Alexandra Vogel[1], Shyam Gopalakrishnan[2], Peter Wad Sackett[1], Gabriel Renaud[1]

1 Department of Health Technology, Technical University of Denmark., Kongens Lyngby, Denmark
2 Section for Hologenomics, University of Copenhagen, Copenhagen, Denmark.

***** jdru@dtu.dk

Abstract

Current mitochondrial DNA (mtDNA) haplogroup classification tools map reads to a single reference genome and perform inference based on the detected mutations to this reference. This approach biases haplogroup assignments towards the reference and prohibits accurate calculations of the uncertainty in assignment. We present `HaploCart`, an mtDNA haplogroup classifier which uses `VG`'s pangenomic reference graph framework together with principles of Bayesian inference. We demonstrate that our approach significantly outperforms available tools by being more robust to lower coverage or incomplete consensus sequences and producing phylogenetically-aware confidence scores that are unbiased towards any haplogroup. `HaploCart` is available both as a command-line tool and through a user-friendly web interface. The program written in `C++` accepts as input consensus FASTA, FASTQ, or GAM files, and outputs a text file with the haplogroup assignments along confidence estimates. Our work considerably reduces the amount of data required to obtain a confident mitochondrial haplogroup assignment. `HaploCart` is available- as a command-line tool at `https://github.com/grenaud/vgan` and as a web server at `https://services.healthtech.dtu.dk/service.php?HaploCart`

Background

The human mitochondrial genome is a small molecule, comprising a mere 16.5 Kb, yet certain properties of the mitogenome make it an invaluable trove of information for researchers in a number of disparate fields including health and human population studies[1]. Mitochondrial DNA (mtDNA) is clonally inherited, meaning that a mother and her child will harbour the same mitogenomic sequence in the absence of mutations. When mutations do occur (which happens more frequently than in the autosomes) a new branch breaks off on the mitochondrial phylogenetic tree. The need to classify the diversity of human mitogenomes has given rise to haplogroups. These are a set of alphanumeric labels which represent monophyletic clusters where individuals in the same haplogroup generally share the same set of idiosyncratic mutations and share a more recent mitochondrial ancestry.

Accurate mtDNA haplogrouping is critical in a number of fields. For one, mitochondrial mutations can affect the phenotype, and there are a number of known associations between mtDNA haplogroup and susceptibility to diseases, such as Parkinson's Disease, Alzheimer's disease, and Chronic Kidney Diseases[2, 3, 4, 5]. Moreover, reliable mtDNA haplogrouping is sometimes applicable in forensic analysis, e.g. for the identification of the deceased in cases of extensive damage[6, 7]. Finally, population geneticists routinely rely on the mitochondrial genome to yield insights into the dynamics of modern and ancient populations[8, 9, 10].

mtDNA haplogrouping methods fall into one of two classes; they either run inference on a consensus sequence, or they perform inference directly on next-generation sequencing (NGS) reads. In the former case, the consensus can be obtained via a consensus-calling program such as `ANGSD`[11] or `mutserve`[12], or via a database of mitogenomes such as the NCBI Nucleotide database. For instance, `Phy-Mer` performs inference on a consensus based on k-mer similarity[13]. `MitoTool` predicts haplogroups based on a consensus on the principle of "optimal exact matching and fuzzy or near matching"[14]. `HaploTracker` ranks haplogroups based on variant identity compared to the canonical Phylotree sequences[15]. Similarly, `HaploGrep2` determines haplogroups by computing the Kulczynski measure between the sample and the putative haplogroup based on sets of expected and observed polymorphisms, weighted by the relative recurrence of each polymorphism in the phylogenetic tree[16].

Only a few programs perform inference directly on NGS reads aligned to a reference. One such program is `mixemt`[17], which performs maximum-likelihood estimation to determine the haplogroups and proportions for each contributor to an mtDNA mixture. The other is `HaploCheck`, a wrapper program which runs `HaploGrep2` on a consensus called upstream (by `mutserve`) for BAM input and performs further downstream analyses[18]. A potential complication with aligning NGS reads directly to a mitochondrial reference is the presence of regions in the nuclear genome that share some similarity with the mitochondrial one. Reads stemming from such regions, otherwise called NuMTs, can align to the mitochondrial reference as well[19].

Several haplogrouping tools are solely available as web servers, e.g. EMMA[20]. Web servers are important for user convenience, but they are simply impractical for high-throughput analysis or large-scale benchmarking experiments. In 2021, a study benchmarked haplogrouping tools on high-throughput sequencing data, among command-line interface (CLI) haplogrouping tools aimed at single-source samples (i.e. the sample is not a mixture of multiple contributors, be they human or contaminant). `HaploCheck` (for BAM input) and `HaploGrep2` (for consensus FASTA input) are the most accurate and robust tool, especially for short reads. For example, `HaploCheck` was the only algorithm to correctly classify all samples in their whole-exome dataset for BAM input and `HaploGrep2` was the only CLI tool to correctly classify all samples in their whole-genome dataset[21]. The conclusion that `HaploGrep2` is currently the most reliable tool reaffirms a view

which had previously been propounded in the literature[22].

Current mtDNA haplogrouping pipelines present a number of shortcomings. For one, programs which call a consensus will typically map reads to a single linear reference genome, quite often to a Eurasian reference known as the rCRS[23]. They are thereby susceptible to reference bias meaning that DNA fragments similar to the linear reference are retained or called with higher quality. This constitutes undesired behavior since the underlying haplogroup of the linear reference is arbitrary[24, 25]. This bias entails that a lack of mutation to the reference may be conflated with missing data due to poor coverage and lead to potentially misleading experimental results. Also, the mitogenome is circular, so mapping to a linear reference may bias against reads which span the artificial rCRS junction. An additional shortcoming of contemporary methods is the absence of sensible reporting of the confidence in haplogroup assignments. For example, we have found from our reported quality experiment that `HaploGrep2` quality scores will always be 0.5 when the predicted haplogroup is that of the rCRS. Finally, while current methods will converge to the correct haplogroup assignment given sufficient coverage depth, their inference lacks a solid mathematical foundation and is therefore ill-suited to sparse data.

We introduce `HaploCart`, a novel maximum-likelihood inference method using genome graphs that significantly outperforms current methods for inferring human haplogroups. We show that our approach is substantially more robust to sparse and incomplete data. As our tool uses a database built on a comprehensive set of human mitogenomes rather than relying on a single one, it fully addresses the biases plaguing current approaches, namely with regards to alignment and the estimation of confidence. Moreover, genome graphs have the ability to be circularized thus solving the issue of linearity in representing mitogenomes. Crucially, we report a phylogenetically-aware confidence score based on the Bayesian theorem that is unbiased towards any haplogroup. Our software, written in C++, takes either unaligned NGS reads in FASTQ format or FASTA consensus. It is released under a GPL v3.0 license. As a rough estimate, haplogroup prediction on a single consensus FASTA sequence takes around thirty seconds using one thread, and around nine seconds using eight threads. `HaploCart` is a `C++` program released as a subcommand of `vgan`, a suite of tools for pangenomics. The program supports multi-threading and is supported for Linux systems.

We demonstrate `HaploCart`'s improvement over current methods in three ways. First, we show an improved ability to call haplogroups on consensus FASTA sequences at varying levels of contiguous masked bases compared to `HaploGrep2` as well as `Phy-Mer`, the only other CLI tool that we know of which calls the haplogroup from a single-source consensus sequence. Then we demonstrate improved predictions, at a higher call rate, for downsampled FASTQ input (both simulated and empirical) compared to `HaploGrep2`. This is true both in terms of the distribution of edit distances between ground truth and predicted haplogroups, and in terms of the total number of predictions that are exactly

correct. Finally, we show qualitatively that HaploCart reported confidence values are more sensible than those of HaploGrep2 with respect to downsampled FASTQ input. Taken together, the experiments indicate that HaploCart predictions and associated reported confidence levels considerably improve current methods for mtDNA haplogroup prediction, showcasing the power of pangenomic reference structures in bioinformatics pipelines.

Results

HaploCart accepts as input either consensus FASTA or FASTQ files. We first present results on consensus FASTA and then we present results on simulated and empirical FASTQ datasets. Finally we discuss our method of sensibly reporting confidence and the computational requirements of the program. For clarity, we will therefore refer to HaploGrep2 when speaking about HaploCheck when used on NGS reads as the underlying algorithm by HaploCheck is HaploGrep2.

Consensus FASTA

Robustness to Missing Data

Missing or unresolved bases, usually denoted by "N"s, in the consensus sequence can be the result of insufficient coverage especially if minimal coverage filters are used. We therefore test the robustness of HaploCart, HaploGrep2 and Phy-Mer to missing bases in consensus FASTA sequences. We took a total of 23 full mitogenomes in FASTA sequences selected from various populations to encapsulate the diversity of human mitogenomes. Details about the sequences that were used for the benchmark are found in the Methods section on page 15. As mentioned in that section, the predicted haplogroups concord on these samples when all bases are unambiguous. Indeed, HaploCart and HaploGrep2 exactly concord on all 23 consensus FASTA sequences (although this is not the case for Phy-Mer, for which two samples are discrepant due to an outdated version of Phylotree).

As missing regions are generally contiguous on mitogenomes due to read length, we masked regions of various length contiguously i.e. a single region is masked on a mitogenome. We masked between a 1Kb to 16Kb bases with a step of 1Kb and created 100 replicates for each number of bases as to measure average behavior.

The results of the masking experiment confirm that HaploCart outperforms HaploGrep2 on input with masked regions, both in terms of the total number of exactly correct predictions (Fig. 1) and in terms of the distribution of edit distances between ground truth and predicted haplogroups for every level of masking from 2Kb up to 16Kb. (Fig. S1). While we see HaploGrep2 narrowly outperforming at the 1Kb level in terms of exact correctness, we actually see the reverse (i.e. HaploCart narrowly outperforming HaploGrep2) in terms of

log edit distance at this level of masking.

Similarly, `HaploCart` outperforms `Phy-Mer` on both metrics from 1Kb to 15Kb of masking. Interestingly, at the 16Kb window `HaploCart` does not perform as well as `Phy-Mer` with regards to log edit distance. This case corresponds to a situation where only about 569 bases are available. This is surprising to us, but perhaps at this extreme level of sequence ambiguity, factors such as mapping sensitivity (in our case, the sensitivity of `vg giraffe` using our selected minimizer index) become the more important factors and `giraffe`'s parameters are not currently tuned for these edge cases. Despite this, our experiment results indicate overall that `HaploCart` is more robust to missing data in the input sample, underscoring the ability of the inference algorithm to gain maximal information from the data available via Bayesian inference.

Empirical Consensus FASTA

To ensure testing on a wide range of potential mitochondrial haplogroups, `HaploCart` and `HaploGrep2` were also run on a dataset of 311 consensus mitogenomic sequences in FASTA format. This dataset has been previously used in several studies involving human mitogenomes [26, 27, 28]. The two programs concorded perfectly on the majority (268 out of 311) of samples (see results and accession IDs in Tab. S4. There were very minor discrepancies in labeling (e.g. `HaploCart`: U6a, `HaploGrep2`: U6a+16189) for 36 samples. These differences are explained by differences in databases between the two programs (i.e. haplogroups with the same name split across two nodes in Phylotree as discussed on page 10). In addition there were some minor differences in assignments for another 5 samples (e.g. `HaploCart`: L2a1a, `HaploGrep2`: L2a1a2) and more substantial differences for 2 out of 311 (e.g. `HaploCart`: T2e1a, `HaploGrep2`: T). These discrepancies are caused by difference in the weighting of mutations between the two inference algorithms. The ground truth haplogroups of these samples are not known so we cannot say which program is correct. Taking this last case as an example, a pairwise alignment of the sample sequence (NCBI accession AF381985) with the canonical sequence of both predicted haplotypes using the EMBOSS Water web server[29] indicates that there are 16558 identical bases (out of a total of 16571) for the `HaploGrep2` prediction and 16559 identical bases for the `HaploCart` prediction. This is a good illustration of why haplogroup assignments in a multi-class classification context can only be viewed as an approximation of the true underlying phylogeny of the sample.

Paired-end FASTQ

Robustness to Low Depth of Coverage

The power of Bayesian modeling is most clearly seen in the ability to gain maximal information from the input data. In cases where data is scant, it is important to ensure that the haplogrouping algorithm makes good use of information that is available. Here we illustrate the ability of `HaploCart` to

robustly assign mtDNA haplogroups on downsampled paired-end FASTQ data. Since `HaploGrep2` and `HaploCheck` run the same algorithm but accept different input formats, in the context of our experimental results, "`HaploGrep2`" refers to the algorithm rather than the program.

We generated a dataset of simulated NGS reads from 23 mitochondrial consensus files in FASTA format for which we know the original haplogroup (see for more details). Briefly, we simulated `Illumina` NGS reads from these 23 consensus sequences using `ART`[30] at read lengths of 50bp and 100bp, and varying target coverage depths while accounting for the circular nature of the mitogenome (further details can be found on page 16). We did so both with and without incorporating reads from nuclear pseudogenes of mitochondrial origin, referred to as NuMTs, a potential confounding source (see 1.2 for more details).

However, simulations do not account for various *in vivo* processes that might create noise that is not present in our simulations. As such, we downsampled a set of eight CRAM files from empirical whole-genome sequencing data from the 1000 Genomes Project. Further details about the downsampling and selection procedure for the CRAM files can be found on page 16.

We found that `HaploCart` significantly outperforms `HaploGrep2` on both simulated and empirical data (Figs. 2 and 3). This is evidenced by the mean edit distance between ground truth and predicted haplogroup, which is always considerably better for `HaploCart` at all coverage depth windows in both experiments. Moreover it can be seen from the shape of the distributions that `HaploCart` predictions tend to have lower variance across replicates within the same depth window.

For the simulated data, the experiment without the added NuMT reads displays nearly identical results with only slightly discernible differences (in both directions) in the edit distances of the most outlying predictions for `HaploCart` (see Fig. S5). This provides evidence that NuMT reads should not constitute a significant impediment to accurate and confident mtDNA haplogrouping.

A similar trend is seen when examining only correct haplogroup assignments rather than distance. `HaploCart` significantly outperforms `HaploGrep2` in total count of precisely correct predictions, on both empirical and simulated data, at greater or equal call rate, for all tested mean depths of coverage (Figs. 4 and S2).

HaploCart Posterior Probabilities are Reasonable Measures of Credence in Predictions

Mitochondrial haplogroups are defined by a characteristic set of mutations in the mitochondrial genome, and the difference between two distinct haplogroup assignments can be as little as a single polymorphism. Therefore when data is sparse it may be theoretically impossible to distinguish between a number of possible haplogroup assignments. The only alternative is to place the sample

within a subset of the mitochondrial tree, with a certain probability. For this reason `HaploCart` reports clade-level posterior probabilities for its haplogroup assignments as a way of reporting confidence in the prediction (see page 11 for more details). Here, we show that these posterior probabilities correlate with coverage depth and are not biased towards any particular human population.

As expected, `HaploCart` posterior probabilities asymptotically approach 1 as the considered clade goes back to the mt-MRCA on both the empirical and simulated paired-end FASTQ datasets (Fig. 5, Figs. S6-S61). Moreover, this convergence to 1 becomes more rapid with a greater depth of coverage, as expected.

Additionally, when we apply a lower threshold to the posterior probability of a sample's haplogroup assignment, we see a clear improvement in the distribution of edit distances in both the masking of consensus sequences in FASTA format and simulated downsampling experiments in FASTQ format (see Figs. S4 and 3 for consensus FASTA and paired-end FASTQ respectively). The degree of improvement is clearly commensurate with the stringency of the threshold. In addition to validating the utility of our posterior probabilities, this demonstrates that quality control measures can easily be applied to experiments which make use of `HaploCart` to reduce the risk of erroneous predictions.

In contrast, `HaploGrep2` quality scores suffer heavily from reference-related issues (Fig. 6). Strikingly, every H2a2a1 replicate with a `HaploGrep2` prediction is associated with a quality score of exactly 0.5, as can be seen in the regression curve on the top-left subplot. This, incidentally, is the quality associated with prediction on the rCRS itself. We have also observed an overwhelming proportion of incorrect `HaploGrep2` predictions being associated with a predicted haplogroup of this reference haplogroup. For example, in our simulated downsampled paired-end FASTQ experiment (without added NuMTs) we found that out of 8849 `HaploGrep2` predictions with an edit distance over 30 from the ground truth haplogroup, 20.08% of the predictions are H2a2a1 (see `score.py` in the data repository on page 18). This constitutes undesired behavior and makes `HaploGrep2` quality scores difficult to interpret.

Since `HaploCart` phylogeny-aware posterior probabilities represent and behave like true probabilities, and since they are agnostic to the haplogroup of the sample, they can be used and interpreted more readily as a measure of confidence in the predictions than `HaploGrep2` quality scores.

Runtime and Peak Memory Usage

`HaploCart` peak memory usage is typically under 2 Gb. For a single consensus FASTA sequence runtime is around thirty seconds on a single thread, and around ten seconds using eight threads (Tables S4, S5, S6). Runtimes tend to improve all the way up to twelve threads or so, after which additional threads do not seem to significantly reduce the runtime (Fig. S7).

Discussion

The adverse effects of reference bias in contemporary mtDNA analysis pipelines are not only present but also widespread and consequential. For example, reference bias effects are known to cause problems in variant calling on ancient DNA[31][32] as well as in allelic expression analysis[33]. Pangenomics is quickly becoming a new paradigm for the analysis of NGS data, allowing for a reduction in reference bias by mapping to an entire collection of genomes simultaneously. The application of pangenomics to the task of mtDNA haplogroup predictions enables significant improvement over the current gold standard, at least among CLI tools. We expect our program to reduce sources of error arising from mtDNA haplogroup calling and potentially reduce sequencing costs by requiring a lower mean depth of coverage per sample.

Despite the demonstrated improvement of `HaploCart` over the current state of the art, `HaploCart` makes a number of simplifying assumptions. For the time being, the program assumes zero contamination from bacterial or exogenous human sources, as well as zero heteroplasmic variants in the sample. Furthermore, we restrict the usage of our algorithm to a single unmixed sample originating from a modern human. Also, when masking the bases of consensus FASTA files, we masked a single region of contiguous bases. In reality, missing regions may comprise a number of missing contiguous segments of varying sizes, a situation for which we do not yet have a good model. All of these assumptions can be addressed by a more sophisticated model, providing opportunities for improvement to the robustness and flexibility of the program.

We envision `HaploCart` being useful for lowering the costs of high-throughput medical experiments requiring a large number of accurately classified samples, for instance in phenome-wide association studies. We also believe that the solid mathematical foundations of the `HaploCart` algorithm lay the groundwork for future extensions and generalizations to handle more difficult samples.

The fact that our program under-performs with respect to `Phy-Mer` in the masking experiment on consensus FASTA sequences (page 4) must be remarked upon. The precise cause of this is unclear but it likely relates to our `giraffe` minimizer index parameters which have not been explicitly tuned to handle this extreme level of masking. `Phy-Mer` uses an alignment-free approach to calling haplogroups, allowing the program to sidestep issues of mapping data at this extreme level of unresolved nucleobases.

We note that `HaploCart` has not been specifically tested for robustness to errors in the consensus sequence, although we see no reason why the algorithm should be more susceptible to these sources of error as compared to unresolved bases.

Given that `HaploCart` is able to confidently infer haplogroups at lower coverage depths than was previously possibly, ancient DNA researchers may wish to use our tool on their data. While we believe the results provided here suggest that our tool may be of use in these cases, we caution that `HaploCart` has not been

properly tested on ancient data, nor can we provide best practices for use of `HaploCart` in this setting.

Finally, it should be noted that the haplogroup assignments in Phylotree or any other man-made tree cannot entirely reflect the true mitochondrial diversity in humans. The refinement of the mitochondrial tree is an ongoing process as more mitogenomes from understudied populations are being sequenced. As this refinement continues `HaploCart` will update its graph accordingly,

Conclusions

Through the reduction of reference bias towards an individual linear mitogenome, a pangenomic approach to human mtDNA haplogroup classification, in conjunction with the power of Bayesian inference, allows for confident and unbiased mtDNA haplogroup assignments of DNA samples even at very low levels of coverage. Unlike the majority of contemporary methods, a Bayesian/pangenomic approach enables precise quantification of the certainty of predictions, conditional on the model outlined above, and lays a solid mathematical foundation for the algorithm which may readily be generalized.

Methods

We first present how our graph was constructed and how we perform inference to predict the mitochondrial haplogroup of the sample. Then we show how test data was generated and how benchmarking experiments were designed.

Graph Construction and Inference

Variation Graph Construction

A variation graph is a bidirected graph embedded with a set of haplogroups such that nodes store DNA segments, edges connect segments that are contiguous along a path, and path sequences can be reconstructed by walking the appropriate nodes in the appropriate orientation.

A variation graph was constructed from our set of haplogroup sequences using the pangenome graph builder (`PGGB`) with parameters $s = 5000, p = 97, n = 5180$[34]. The resultant graph in GFAv1 format comprises a single connected component with 11821 nodes and 16245 edges, and 5179 embedded paths, one per haplogroup[1][35]. The graph was converted to `VG` format (a serialized graph format using protocol buffers) with `vg convert`, into nodes with a maximum sequence size of 8 bases per node using `vg mod -X`, and circularized with `vg circularize` to reflect the circular nature of mitochondrial DNA. In addition to

[1]There are five instances of haplogroups in Phylotree defined by an unspecified number of inserted cytosines in a poly-C tract. In these instances we do not consider the insertions to be informative, and do not differentiate these haplogroups from their parent haplogroups.

the graph topology we also circularized the embedded haplogroups with a custom script (`circularize_paths.py` in the data repository (see page 18). Finally the circularized `VG` graph was converted to PathHandleGraph[36] format (`vg view -o`) to make it easier to read into memory at runtime.

Mapping `HaploCart` accepts as input FASTA, FASTQ (single or paired-end[2]), and GAM (`VG`'s graph analog of BAM). For all input file formats (except GAM) mapping is performed through `vg giraffe` using internal C++ function calls, which maps to k-mers arising from the graph's embedded haplogroups[37]. For FASTQ input, mapping is conducted under "fast" mode for increased performance. `giraffe` maps using an index of special k-mers called minimizers (because they minimize a certain objective function within a given window)[38]. We use a minimizer index with window size of 11 and k-mer size 31, unless there are many ambiguous bases detected, in which case we use a more sensitive minimizer index with a smaller k-mer size. In the case of consensus FASTA input, we generate synthetic FASTQ reads with dummy quality scores correspondent with the background error probability in the consensus call. Since `giraffe` does not index k-mers with ambiguous bases, we replace such bases with adenines (arbitrarily) of quality zero.

As long-read technologies become more ubiquitous we anticipate that `vg giraffe` will be furthered developed to handle mapping longer reads to embedded paths, and indeed the `VG` team is actively working on this (`https://github.com/vgteam/vg/wiki/Roadmap`). When this happens `HaploCart` may be able to map consensus FASTA directly and bypass the FASTA-to-FASTQ conversion step.

After mapping, unmapped reads and reads with overhangs (inserts at either flank of a read) are discarded with `vg filter`. The resulting GAM file is sorted with `vg gamsort` and PCR duplicates are removed as described on page 1.

Embedded Haplogroups One synthetic FASTA file was constructed for each named haplogroup assignment in Phylotree build 17. Briefly, we wrote a custom script to convert a set of variants to HSD format, and a secondary script (`write_hsd.py` in the data repository) to convert HSD to FASTA[39]. We verified through spot-checking that `HaploGrep2` predictions on these synthetic sequences exactly match all the labelled haplogroups, with all expected polymorphisms present and zero private mutations found. These sequences were then compiled into a multifasta file. Further details about the resulting graph can be found on page 9.

Phylotree Build 17 Phylotree is a hand-curated human mitochondrial tree that is very widely used in the mtDNA research community[40]. Briefly, it contains 5437 nodes with 5179 named haplogroups.

[2]Due to the highly collapsed topology of the graph, `giraffe` in its present version seems to always fall back to single-end mapping even with paired-end data; however, we believe the adversely impact of this is negligible.

Although the most common, Phylotree is not the only mtDNA tree used by researchers. By default, `HaploGrep2` uses a more refined tree by Dür *et. al.* with an increased number of haplogroup-defining motifs[41]. Phylotree Build 17 was selected to build our graph, rather than the Dür *et. al.* tree, because it is more widely accepted within the mtDNA community — to our knowledge `HaploGrep2` is the only program which makes use of this tree. For instance, it is impossible to use the Dür *et. al.* tree while running `HaploCheck` on empirical BAM samples, despite the fact that `HaploCheck` uses `HaploGrep2` for haplogroup classification.

In 258 (5437 nodes - 5179 named haplogroups) cases a named haplogroup is divided among two nodes in Phylotree, being differentiated by a single polymorphism. In these cases `HaploCart` does not currently distinguish between the two nodes, considering them to belong to the same haplogroup assignment We believe this is justified because in these cases there is no agreed-upon nomenclature to distinguish the two nodes, so it is more sensible to view them as two subpopulations of the same haplogroup assignment.

To parse the tree structure of Phylotree (for posterior calculations) we use a modified version of a script from the `mixemt` program repository[17]. In particular we modified their script `phylotree.py` into a script which produces three TSV files, one file listing the parent clades to each haplogroup assignment, one file listing the children clades to each haplogroup assignment, and one file listing the defining polymorphisms of each haplogroup assignment. Our modified script `children_parents.py` can be found in the data repository on Github (see page 18).

The `HaploCart` algorithm is not reliant on any particular tree and we expect future versions to support a number of reference trees, potentially including reconstructed ancestral sequences.

Inference

A graphical overview of the inference algorithm writ large is provided in Fig. 7. Here we provide the mathematical details.

Inference is performed under a maximum-likelihood framework. Let $\mathcal{H}$ denote our embedded set of haplogroups in the variation graph. Given a set of DNA reads $\mathcal{R}$, the goal is to find the haplogroup h_k maximizing $P(h_k|\mathcal{R})$. Bayes' Theorem tell us

$$P(h_k|\mathcal{R}) = \frac{P(\mathcal{R}|h_k)P(h_k)}{P(\mathcal{R})}$$

To mitigate bias against any particular population we employ a uniform prior over all haplogroups in the graph. We remove PCR duplicates as detailed in the Supplementary Methods (page 1). After PCR duplicate removal we can assume

reads are independent, so we can take the product of probabilities conditional on each read r_i

$$P[\mathcal{R}|h_k] = \prod_{r_i \in \mathcal{R}} P[r_i|h_k]$$

Furthermore, since mitochondrial DNA is clonally inherited and no recombinations occurs, we can assume observations of bases are independent. Thus we can further decompose this term into a product over individual bases observed $b_{i,j}$ observed in read r_i

$$P[r_i|h_k] = \prod_{b_{i,j} \in r_i} P[b_{i,j}|h_k]$$

Let M denote the event that a given read is correctly mapped, while $\neg M$ is the event that the read is incorrectly mapped. Then we can express this probability as the sum of two disjoint events:

$$P[b_{i,j}|h_k] = P(\neg M)P[b_{i,j}|h_k, \neg M] + P(M)P[b_{i,j}|h_k, M]$$

The event $\neg M$ also covers cases in which the read is derived from NuMTs. NuMTs are nuclear pseudogenes of mitochondrial origin which can be difficult to distinguish from genuine mtDNA and can therefore hinder accurate haplogrouping[42]. To mitigate the adverse effect of potential NuMTs in the sample, we assigned weights to each mitogenomic base proportional to the mappability of that base across the nuclear genome (see page 1 in the Supplementary methods for more details).

If the event $\neg M$ has occurred (i.e. the read is incorrectly mapped) the probability of observing a base of the considered read given that the read arose from the putative haplogroup is entirely independent from the base in the graph for this haplogroup, as the alignment is spurious. The probability of observing the base is therefore dictated by the background frequency of bases in the graph (Tab. S3).

$$P[b_{i,j}|h_k, \neg M] = \pi(b_{i,j})$$

In the event that the DNA fragment is correctly mapped, we can therefore trust the alignment and any substitution from the haplotype h_k in the graph is due to either mutation or sequencing error. However, we do not know the base of the mitochondrial genome from the mitochondrial sample being analyzed. Let us denote the base present in the mitochondrial sample as g. It is possible that

a mutation might have occurred between the one from h_k and the mitochondrial genome of the sample. Therefore, we marginalize over the four possible nucleotides for $b_{i,j}$ as such:

$$P[b_{i,j}|h_k, M] = \sum_{g \in \{A,C,G,T\}} P[h_k \to g] P[b_{i,j}|g]$$

where $P[h_k \to g]$ denotes the probability that the sample is haplogroup h_k and harbors base g at the position in question. This probability is given by

$$P[h_k \to g] = \begin{cases} (1-\mu)^{generation} & b_{i,j} = g \\ \left(1-(1-\mu)^{generation}\right) * \frac{1}{46} & \text{transversion} \\ \left(1-(1-\mu)^{generation}\right) * \frac{22}{23} & \text{transition} \end{cases}$$

where μ is the site-specific mutation rate at the considered mitogenomic position, and $generations$ is the number of generations between the sample and the emergence of the putative haplogroup. We set $generations = 8$ as an arbitrary hyperparameter since this value is aleatorically uncertain and will vary from sample to sample. The appropriate site-specific mutation rate is determined via a precomputed annotation using the `position` command from `ODGI`[43] to surject graph coordinates (nodes, not individual bases) onto the H2a2a1 haplogroup path. In other words, the H2a2a1 path provides a pangenomic coordinate system for the graph, so that each node in the graph maps on to a base in this "reference" path. We use maximum-likelihood-derived estimates of μ from the literature[44]. For simplicity, in protein-coding regions we take a weighted average of the mutation rates for first or second position and final position of the codon. The transition/transversion rate $\frac{ti}{tv}$ in human mtDNA is approximately $\frac{1}{22}$[45]. We distribute the probability mass of a mutation accordingly.

It is possible that an aligned base is on a node which does not support the haplogroup being evaluated, i.e. the walk through the putative haplogroup never traverses this node. To illustrate, say a reads aligns to a node containing a "C" whereas the node of the haplogroup whose likelihood is being computed contains a "T" on a different node. In this case, that haplogroup incurs a penalty commensurate with $\frac{3}{4}$ mismatch and $\frac{1}{4}$ match of the node's sequence at the given base quality scores[3].

As we are assuming that the sample harbors base g at position $b_{i,j}$, there are two possibilities for what base is observed. Either the base is correctly called by the sequencing machine, or else a sequencing error has occurred. Thus the probability of observing base $b_{i,j}$ is dictated by the quality score for that base,

[3]We note that the penalty for an untraversed node is somewhat arbitrary. In theory this penalty can be learned, but we have found that it has little, if any, affect on inference, so long as it is sufficiently harsh.

$$P[b_{i,j}|g] = \begin{cases} 1 - \epsilon_{b_{i,j}} & b_{i,j} = g \\ \epsilon_{b_{i,j}} & b_{i,j} \neq g \end{cases}$$

Here $\epsilon_{b_{i,j}}$ is the probability that a sequencing error has occurred which is computed from the base quality scores according to the usual PHRED scale encoding. For speed, these are precomputed by the program. If no quality scores are provided (i.e. for FASTA input) we employ a fixed background error probability (default 0.0001).

The computation of $P[b_{i,j}|h_k]$ is graphically illustrated in Fig. 8.

Posterior Probabilities Sometimes, for very low coverage data for instance, it may be important to know not just the predicted haplogroup but also an estimate of the confidence in the prediction. For these cases `HaploCart` optionally performs phylogeny-aware posterior confidence estimation for the predicted haplogroup assignment, as well as for each "depth", i.e. ontological level up the tree.

As an example, suppose that `HaploCart`'s most likely assignment is Z3. Recall that

$$P(\text{`}Z3\text{'}|\mathcal{R}) = \frac{P(\mathcal{R}|\text{`}Z3\text{'})P(\text{`}Z3\text{'})}{P(\mathcal{R})}$$

This is the posterior probability that the underlying haplogroup assignment is precisely Z3. The parent haplogroup to Z3 is haplogroup Z which is within the CZ family which itself falls within the M superhaplogroup which in turn is within the L3 superhaplogroup. If we want to know the total posterior probability mass that falls into the subtree with MRCA Z3 (call it $\widehat{Z3}$), we simply sum the posteriors over the set of haplogroups within the appropriate clade, i.e.

$$P(\widehat{Z3}|\mathcal{R}) = \sum_{h \subseteq Z3} \frac{P(\mathcal{R}|h)P(h)}{P(\mathcal{R})}$$

Where $X \subseteq Y$ means that haplogroup X falls within the clade derived from the ancestral haplogroup Y. When computing clade-level posterior probabilities `HaploCart` will compute this sum for larger and larger subtrees. For example, if the predicted haplogroup is Z3, then `HaploCart` will report the posterior probability of the sample belonging to Z3, but also to any haplogroup within the parent clade Z, the parent clade to the parent clade CZ, the parent clade to the parent clade to the parent clade M8, and so on. The program assumes the sample originates from a modern human, so the posterior probability of the most ancestral clade (mt-MRCA) is reported to be 1 irrespective of the input.

It is important to note that the reported confidence values implicitly assume that the input sample belongs to one of the haplogroups embedded in the graph,

i.e. in Phylotree build 17. These posterior values do not consider the possibility that the underlying haplogroup is out of model. This is why we have observed that, given a sufficient amount of data, the posteriors will approach an equal confidence among N possible haplogroup assignments for some integer N.

Benchmarking `HaploCart` Performance

We tested `HaploCart` on both *in silico* and empirical data. Ground truth haplogroup assignments were obtained by either running `HaploGrep2` directly on the consensus sequences for simulated FASTA data or via running `HaploGrep2` at full coverage for empirical data.

Data Generation

Selection of mitochondrial genomes for benchmarking We selected 25 consensus FASTA samples from the NCBI Nucleotide Database[46] (see Table 1 for haplogroups and accessions numbers). To ensure independence, these sequences were not embedded in our graph of haplogroups. We attempted to cover a wide range of mitodiversity such that every major clade has some representation to ensure that our results are sufficiently generalized. These FASTA consensus were used both for the FASTA masking experiment and the simulated NGS data in FASTQ format. The `HaploCart` and `HaploGrep2` predictions exactly concorded on 24 of the 25 consensus FASTA input.

The one discordant sample was NCBI accession MN894780, which `HaploCart` predicted to be haplogroup L1c2 and `HaploGrep2` predicted to be haplogroup L1c2b. These two haplogroups differ by three polymorphisms: A11164G, (T16093C), and (G16286A), where the two mutations in parentheses are considered by Phylotree to be unstable/recurrent or uncertain. The sample in question only harbors one of the three mutations. Therefore one cannot say for certain whether this sample was an L1c2 with an extra mutation, or an L1c2b with two back-mutations. Due to this ambiguity in the underlying haplogroup we decided to exclude this sample from our benchmarking.

We further excluded the sample JX154035 of haplogroup H2a2a1 from the benchmarking experiments because we had found that its ground truth haplogroup, H2a2a1, is a prediction the HaploGrep2 algorithm reverts to at low levels of certainty since this is the haplogroup assignment of the mitochondrial reference genome. Including this sample would therefore artificially inflate the performance of `HaploGrep2`. We therefore had a total of 23 consensus FASTA sequences which were used for the masked FASTA experiment and the simulated NGS data in FASTQ format.

However the sample JX15403 was still used in Figure 6 because it is important to demonstrate the effect of reference bias on `HaploGrep2` quality scores.

Consensus FASTA Sequences We investigated robustness to missing data by masking certain bases in the input as to mimic missing data due to lack of breath of coverage or minimal coverage filters being applied for low-coverage samples. As input data, we used the 23 original consensus FASTA sequences described in the section Selection of mitochondrial genomes for benchmarking. For each multiple of one thousand $N \in [1000, 16000]$, we masked an arbitrary contiguous region of N bases in the consensus sequence (potentially spanning the rCRS junction) for each of 100 replicates. This procedure was done for the 23 consensus mitogenomes in FASTA format.

Paired-end FASTQ Here we describe the procedure for generation of both empirical and simulated paired-end FASTQ samples. Generation of replicates for both the masking and the downsampling experiments was automated with `Snakemake` [47].

Simulated Paired-end FASTQ We generated paired-end FASTQ simulations from the 23 mitochondrial consensus files in FASTA format for which we know the original haplogroup (see the section Selection of mitochondrial genomes for benchmarking on page 15). From these consensus sequences, circular fragments of length 125bp were generated with `fragSim` from the `gargammel` using the `--circ` flag[4] program[48]. These circular fragments were used to generate synthetic paired-end Illumina reads (HiSeq 2500 "HS25" sequencing system) were generated with `ART`[49] version 2.5.8 at read lengths of 50bp and 100 bp. Subsampling was performed using `seqtk` version 1.3-r106 at target depths of 0.03X to 0.1X with a step size of 0.1X, as well as 0.2X and 0.3X [50]. 100 replicates were generated per read length per target coverage depth. This workflow was managed using `Snakemake` and the exact commands can be found in the Snakefile (see page 18).

For our downsampling experiment on simulated data we interleaved the files with the external script `interleave_fastq.sh`[51] and passed interleaved FASTQ files directly to `HaploCart`. Since neither `HaploGrep2` nor `HaploCheck` accepts FASTQ input, we first map interleaved FASTQ files to the rCRS using `bwa mem` version 0.7.17-r1188 under default parameters, and then pass the BAM files to `HaploCheck`[52]. This was done in the script `benchmark.py` in the data repository (page 18).

Empirical Paired-end FASTQ In addition to simulated data we also benchmark `HaploCart` performance on empirical paired-end whole-genome shotgun data. This data comprised eight Thousand Genomes Project samples from distinct sequencing centers and were selected so as to capture a wide array of human mitodiversity (this is why Africa is intentionally over-represented)[53] (see Table S2 for samples that were used). The data were downloaded as CRAM files from the FTP server at `http://ftp.1000genomes.ebi.ac.uk/vol1/ftp/data_collections/1000_genomes_project/data`.

[4] GitHub commit: 33de7225447f3f6ed014a674deef3191d5da57df

Subsampling on these raw CRAM files was performed across all chromosomes (both nuclear and mitochondrial) with the `view` command from `samtools` using subsampling rates from 0.1 to 1.0 inclusive at a step size of 0.1[54]. This was done from the `subsample.py` script in the data repository (page 18).

Experimental Setup

The metric used for scoring in all experiments was edit (Levenshtein) distance, a standard measure of distance between genomic sequences. In all cases `HaploGrep2 classify` is run with the `--phylotree 17` flag. If the `HaploGrep2` prediction was "mt-MRCA" we discarded the sample for scoring purposes. A very small number of `Phy-Mer` predictions are not in Phylotree build 17 since the program uses an outdated build. In these cases we discard the samples from our experiment. Also, occasionally `Phy-Mer` will provide multiple predictions, ranked in order of confidence. In these cases we take the top prediction.

For certain applications it may be important to know how well `HaploCart` can pinpoint the precise haplotypic assignment of the sample, down to very fine granularity. Therefore, in addition to examining the distribution of edit distances between ground truth and predicted haplogroups, we also examined the total number of samples that were precisely identified by the respective programs in both the masking experiment on consensus FASTA data and the downsampling experiments on simulated and empirical paired-end FASTQ data.

Robustness to Depth of Coverage in Simulated and Empirical Paired-end FASTQ Data For the downsampling experiment we benchmark against the `HaploGrep2` algorithm. Although `Phy-Mer` purports to run on BAM files, we did not include the program in this experiment since we were unable to get it to run.

This procedure was repeated on the simulated dataset with the inclusion of NuMT sequences at a rate of one NuMT per 200 mitochondrial reads to gauge the efficacy of our NuMT model and see how robust the program is to noise from non-source contributors (See page 1 in the Supplemental Methods for more details).

Software Versions For all experiments we used `HaploCheck` version 1.3.2 and `HaploGrep2` version 2.4.0. All experiments were performed using `VG` version `Obolo`. All `samtools` commands used version 1.13. Our Snakemake version was 5.10.0. Our `ART` version was 2.5.8. Our `ODGI` version was v0.6.3-52-g0d7f950 ("Pulizia").

Declarations

Ethics approval and consent to participate

Not applicable.

Consent for Publication

Not applicable.

Availability of Data and Materials

The datasets generated and/or analysed during the current study are available in the following repository: `https://github.com/JoshuaDanielRubin/HaploCart_Experiment_Data`

Competing Interests

The authors declare that they have no competing interests

Funding

Funding for this research was provided by a Novo Nordisk Data Science Investigator grant number NNF20OC0062491, which provides for the PhD scholarships of JDR and NAV. We also would like to thank the department of Healthtech at DTU for additional funding.

Authors' Contributions

JDR and GR developed and implemented the program. JDR tested the program. NAV and SG provided valuable insights for the testing and interpretation of the data. PWS provided critical IT infrastructure support for the project. All authors read and approved the final manuscript.

Acknowledgements

We would like to thank Daniel Caleb Remero Yianni for his help with the web application and maintenance of computational infrastructure. We would also like to that Viviane Slon and Ana Duggan for their valuable comments on the manuscript.

References

[1] Kivisild T. Maternal ancestry and population history from whole mitochondrial genomes. Investigative Genetics. 2015;6(1):1–10.

[2] Guo JH, Shi JM, Shi GP, Wang Y, Chu XF, Wang ZD, et al. Association Study of Mitochondrial DNA Haplogroup D and C5178A Polymorphisms with Chronic Kidney Disease. Genetic Testing and Molecular Biomarkers. 2021;25(8):546–550.

[3] Pyle A, Foltynie T, Tiangyou W, Lambert C, Keers SM, Allcock LM, et al. Mitochondrial DNA haplogroup cluster UKJT reduces the risk of PD. Annals of Neurology. 2005;57(4):564–567.

[4] Chinnery P, Taylor G, Howell N, Andrews R, Morris C, Taylor R, et al. Mitochondrial DNA haplogroups and susceptibility to AD and dementia with Lewy bodies. Neurology. 2000;55(2):302–304.

[5] Ghezzi D, Marelli C, Achilli A, Goldwurm S, Pezzoli G, Barone P, et al. Mitochondrial DNA haplogroup K is associated with a lower risk of Parkinson's disease in Italians. European Journal of Human Genetics. 2005;13(6):748–752.

[6] Budowle B, Allard MW, Wilson MR, Chakraborty R. Forensics and mitochondrial DNA. Annual Review of Genomics and Human Genetics. 2003;4:119–141.

[7] Emery M, Bolhofner K, Ghafoor S, Winingear S, Buikstra J, Fulginiti L, et al. Whole mitochondrial genomes assembled from thermally altered forensic bones and teeth. Forensic Science International: Genetics. 2022;56:102610.

[8] Finnilä S, Lehtonen MS, Majamaa K. Phylogenetic network for European mtDNA. The American Journal of Human Genetics. 2001;68(6):1475–1484.

[9] Soares P, Alshamali F, Pereira JB, Fernandes V, Silva NM, Afonso C, et al. The expansion of mtDNA haplogroup L3 within and out of Africa. Molecular Biology and Evolution. 2012;29(3):915–927.

[10] Maca-Meyer N, Arnay M, Rando JC, Flores C, González AM, Cabrera VM, et al. Ancient mtDNA analysis and the origin of the Guanches. European Journal of Human Genetics. 2004;12(2):155–162.

[11] Korneliussen TS, Albrechtsen A, Nielsen R. ANGSD: analysis of next generation sequencing data. BMC Bioinformatics. 2014;15(1):1–13.

[12] Weissensteiner H, Forer L, Fendt L, Kheirkhah A, Salas A, Kronenberg F, et al. Contamination detection in sequencing studies using the mitochondrial phylogeny. Genome Research. 2021;31(2):309–316.

[13] Navarro-Gomez D, Leipzig J, Shen L, Lott M, Stassen AP, Wallace DC, et al. Phy-Mer: a novel alignment-free and reference-independent mitochondrial haplogroup classifier. Bioinformatics. 2015;31(8):1310–1312.

[14] Fan L, Yao YG. MitoTool: a web server for the analysis and retrieval of human mitochondrial DNA sequence variations. Mitochondrion. 2011;11(2):351–356.

[15] Kim K, Kim Dh, Kim Ky. Mitochondrial Haplogroup Classification of Ancient DNA Samples Using Haplotracker. BioMed Research International. 2022;2022.

[16] Weissensteiner H, Pacher D, Kloss-Brandstätter A, Forer L, Specht G, Bandelt HJ, et al. HaploGrep 2: mitochondrial haplogroup classification in the era of high-throughput sequencing. Nucleic Acids Research. 2016;44(W1):W58–W63.

[17] Vohr SH, Gordon R, Eizenga JM, Erlich HA, Calloway CD, Green RE. A phylogenetic approach for haplotype analysis of sequence data from complex mitochondrial mixtures. Forensic Science International: Genetics. 2017;30:93–105.

[18] Weissensteiner H, Forer L, Fendt L, Kheirkhah A, Salas A, Kronenberg F, et al. Contamination detection in sequencing studies using the mitochondrial phylogeny. Genome Research. 2021;31(2):309–316.

[19] Gaziev A, Shaikhaev G. Nuclear mitochondrial pseudogenes. Molecular Biology. 2010;44(3):358–368.

[20] Röck AW, Dür A, Van Oven M, Parson W. Concept for estimating mitochondrial DNA haplogroups using a maximum likelihood approach (EMMA). Forensic Science International: Genetics. 2013;7(6):601–609.

[21] García-Olivares V, Muñoz-Barrera A, Lorenzo-Salazar JM, Zaragoza-Trello C, Rubio-Rodríguez LA, Díaz-de Usera A, et al. A benchmarking of human mitochondrial DNA haplogroup classifiers from whole-genome and whole-exome sequence data. Scientific Reports. 2021;11(1):1–11.

[22] Bandelt HJ, van Oven M, Salas A. Haplogrouping mitochondrial DNA sequences in legal medicine/forensic genetics. International Journal of Legal Medicine. 2012;126(6):901–916.

[23] Andrews RM, Kubacka I, Chinnery PF, Lightowlers RN, Turnbull DM, Howell N. Reanalysis and revision of the Cambridge reference sequence for human mitochondrial DNA. Nature Genetics. 1999;23(2):147–147.

[24] Brandt DY, Aguiar VR, Bitarello BD, Nunes K, Goudet J, Meyer D. Mapping bias overestimates reference allele frequencies at the HLA genes in the 1000 genomes project phase I data. G3: Genes, Genomes, Genetics. 2015;5(5):931–941.

[25] Garrison E, Sirén J, Novak AM, Hickey G, Eizenga JM, Dawson ET, et al. Variation graph toolkit improves read mapping by representing genetic variation in the reference. Nature Biotechnology. 2018;36(9):875–879.

[26] Fu Q, Mittnik A, Johnson PL, Bos K, Lari M, Bollongino R, et al. A revised timescale for human evolution based on ancient mitochondrial genomes. Current Biology. 2013;23(7):553–559.

[27] Morris AG, Heinze A, Chan EK, Smith AB, Hayes VM. First ancient mitochondrial human genome from a prepastoralist southern African. Genome Biology and Evolution. 2014;6(10):2647–2653.

[28] Posth C, Wißing C, Kitagawa K, Pagani L, van Holstein L, Racimo F, et al. Deeply divergent archaic mitochondrial genome provides lower time boundary for African gene flow into Neanderthals. Nature Communications. 2017;8(1):1–9.

[29] EMBOSS Water. EMBL-EBI Hinxton;. `https://www.ebi.ac.uk/Tools/psa/emboss_water/`.

[30] Huang W, Li L, Myers JR, Marth GT. ART: a next-generation sequencing read simulator. Bioinformatics. 2012;28(4):593–594.

[31] Martiniano R, Garrison E, Jones ER, Manica A, Durbin R. Removing reference bias and improving indel calling in ancient DNA data analysis by mapping to a sequence variation graph. Genome biology. 2020;21(1):1–18.

[32] Zhou B, Wen S, Wang L, Jin L, Li H, Zhang H. AntCaller: an accurate variant caller incorporating ancient DNA damage. Molecular Genetics and Genomics. 2017;292(6):1419–1430.

[33] Castel SE, Levy-Moonshine A, Mohammadi P, Banks E, Lappalainen T. Tools and best practices for data processing in allelic expression analysis. Genome Biology. 2015;16(1):1–12.

[34] PGGB;. Available from: `https://github.com/pangenome/pggb`. Accessed: 2022-08-24.

[35] Nurk S. GFA: Graphical Fragment Assembly (GFA) Format Specification. GitHub; 2022. `https://github.com/GFA-spec/GFA-spec`.

[36] Eizenga JM, Novak AM, Kobayashi E, Villani F, Cisar C, Heumos S, et al. Efficient dynamic variation graphs. Bioinformatics. 2021;36(21):5139–5144.

[37] Sirén J, Monlong J, Chang X, Novak AM, Eizenga JM, Markello C, et al. Pangenomics enables genotyping of known structural variants in 5202 diverse genomes. Science. 2021;374(6574):abg8871.

[38] Roberts M, Hayes W, Hunt BR, Mount SM, Yorke JA. Reducing storage requirements for biological sequence comparison. Bioinformatics. 2004;20(18):3363–3369.

[39] HaploGrep2 README.md;. Accessed: 2022-08-24. `https://github.com/seppinho/haplogrep-cmd/blob/master/README.md`.

[40] Van Oven M, Kayser M. Updated comprehensive phylogenetic tree of global human mitochondrial DNA variation. Human Mutation. 2009;30(2):E386–E394.

[41] Dür A, Huber N, Parson W. Fine-Tuning phylogenetic alignment and haplogrouping of mtDNA sequences. International Journal of Molecular Sciences. 2021;22(11):5747.

[42] Lopez JV, Cevario S, O'Brien SJ. Complete nucleotide sequences of the domestic cat (Felis catus) mitochondrial genome and a transposed mtDNA tandem repeat (Numt) in the nuclear genome. Genomics. 1996;33(2):229–246.

[43] Guarracino A, Heumos S, Nahnsen S, Prins P, Garrison E. ODGI: understanding pangenome graphs. bioRxiv. 2021;.

[44] Soares P, Ermini L, Thomson N, Mormina M, Rito T, Röhl A, et al. Correcting for purifying selection: an improved human mitochondrial molecular clock. The American Journal of Human Genetics. 2009;84(6):740–759.

[45] Levinstein Hallak K, Tzur S, Rosset S. Big data analysis of human mitochondrial DNA substitution models: a regression approach. BMC Genomics. 2018;19(1):1–13.

[46] NCBI Nucleotide Database;. Available from: `https://www.ncbi.nlm.nih.gov/nucleotide/`.

[47] Köster J, Rahmann S. Snakemake—a scalable bioinformatics workflow engine. Bioinformatics. 2012;28(19):2520–2522.

[48] Renaud G, Hanghøj K, Willerslev E, Orlando L. gargammel: a sequence simulator for ancient DNA. Bioinformatics. 2017;33(4):577–579.

[49] Huang W, Li L, Myers JR, Marth GT. ART: a next-generation sequencing read simulator. Bioinformatics. 2012;28(4):593–594.

[50] Lh3/SEQTK: Toolkit for processing sequences in FASTA/Q Formats;. Available from: `https://github.com/lh3/seqtk`. Accessed: 2022-08-24.

[51] Watson-Haigh NS. `interleave_fastq.sh`. GitHub;. `https://gist.github.com/nathanhaigh/4544979`.

[52] Li H. Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. arXiv preprint arXiv:13033997. 2013;.

[53] Consortium GP, et al. A global reference for human genetic variation. Nature. 2015;526(7571):68.

[54] Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, et al. The sequence alignment/map format and SAMtools. Bioinformatics. 2009;25(16):2078–2079.

Figures

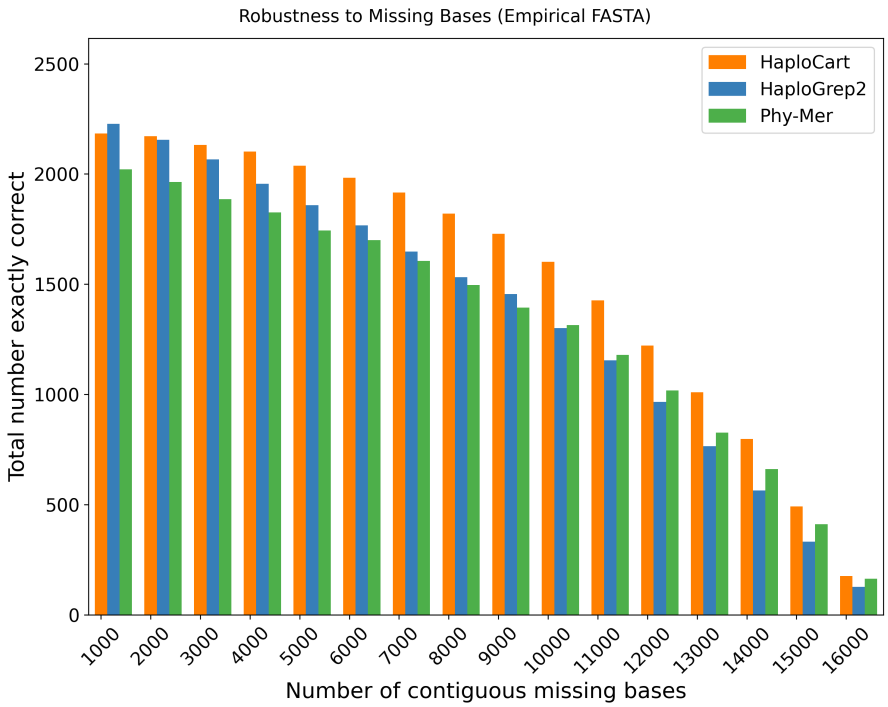

Figure 1: Total Number of Exactly Correct Predictions as a Function of the Number of Contiguous Masked Bases on Consensus FASTA Input. Counts are provided for `HaploCart`, `Phy-Mer`, and `HaploGrep2`. `HaploCart` outperforms the other two programs from 2Kb up to 16Kb.

.

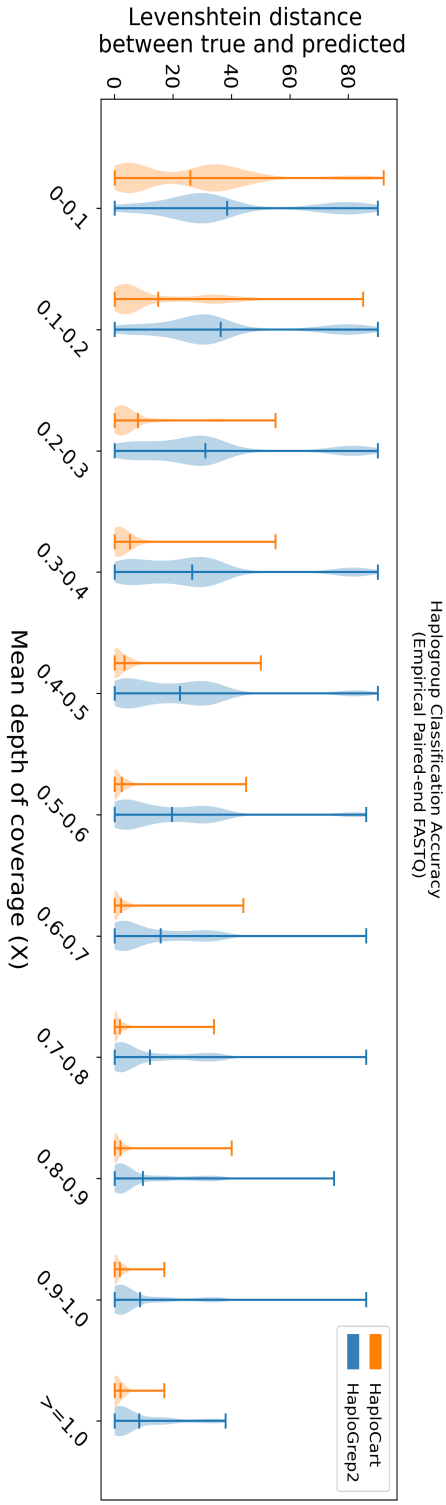

Figure 2: Performance on Empirical Paired-end FASTQ Data. Distribution of edit (Levenshtein) distances between assigned and underlying haplogroup of replicates from the empirical dataset. Central bars represent the arithmetic means of the distribution. For each window, `HaploCart` outperforms `HaploGrep2` as determined by the means of the distribution. It is worth noting that unlike `HaploGrep2`, `HaploCart` makes a prediction if even a single read maps to the graph.

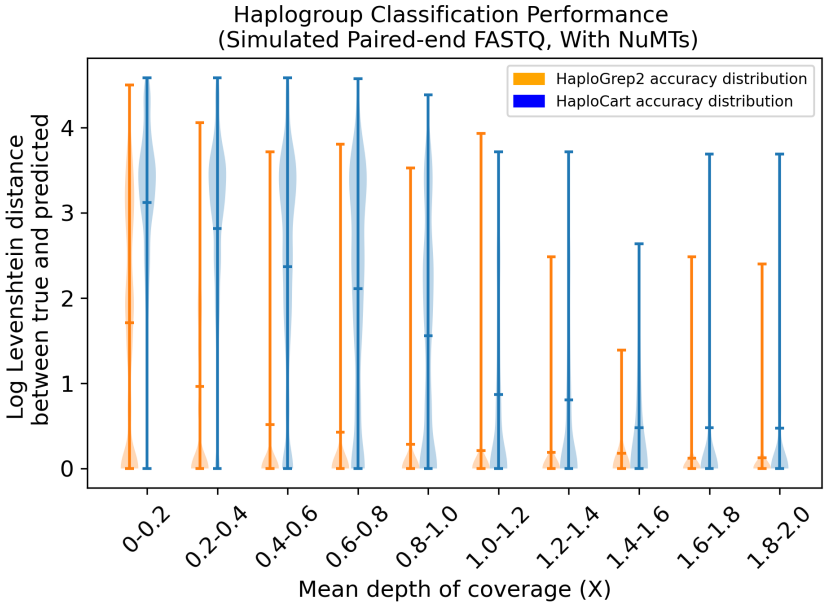

Figure 3: Downsampling Experiment on Simulated Paired-end FASTQ Data with Added NuMT reads at a Rate of One in Two Hundred. Distribution of log Levenshtein (edit) distances between ground truth and predicted haplogroups on simulated paired-end reads with added simulated NuMT reads at a frequency of one in two hundred. Central bars indicate arithmetic means and are considerably lower for `HaploCart` over all windows.

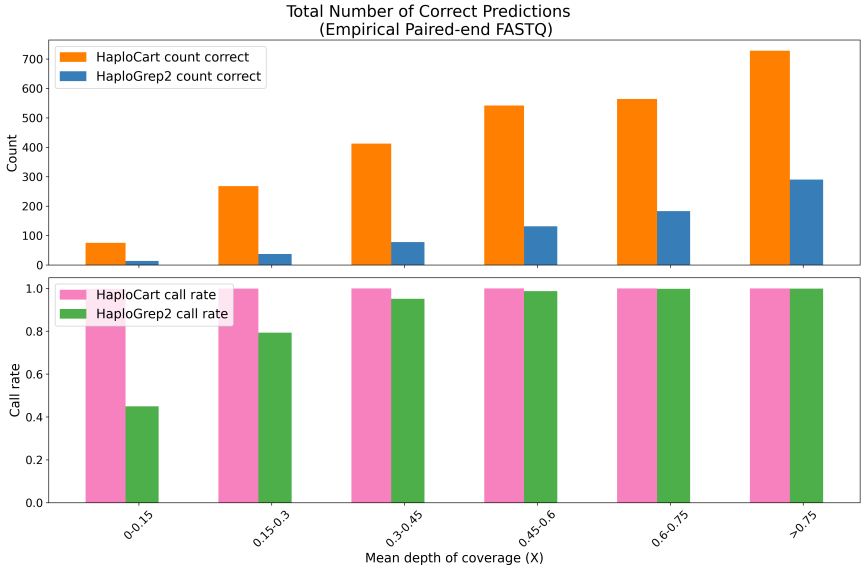

Figure 4: Correctness of Predictions on Empirical Paired-end FASTQ Data. Total count of predictions on the Thousand Genomes Project subsampled replicates which exactly match the underlying haplogroup, as determined by running `HaploGrep2` at full coverage. For each window, `HaploCart` outperforms `HaploGrep2` by providing more reliable haplogroup assignments.

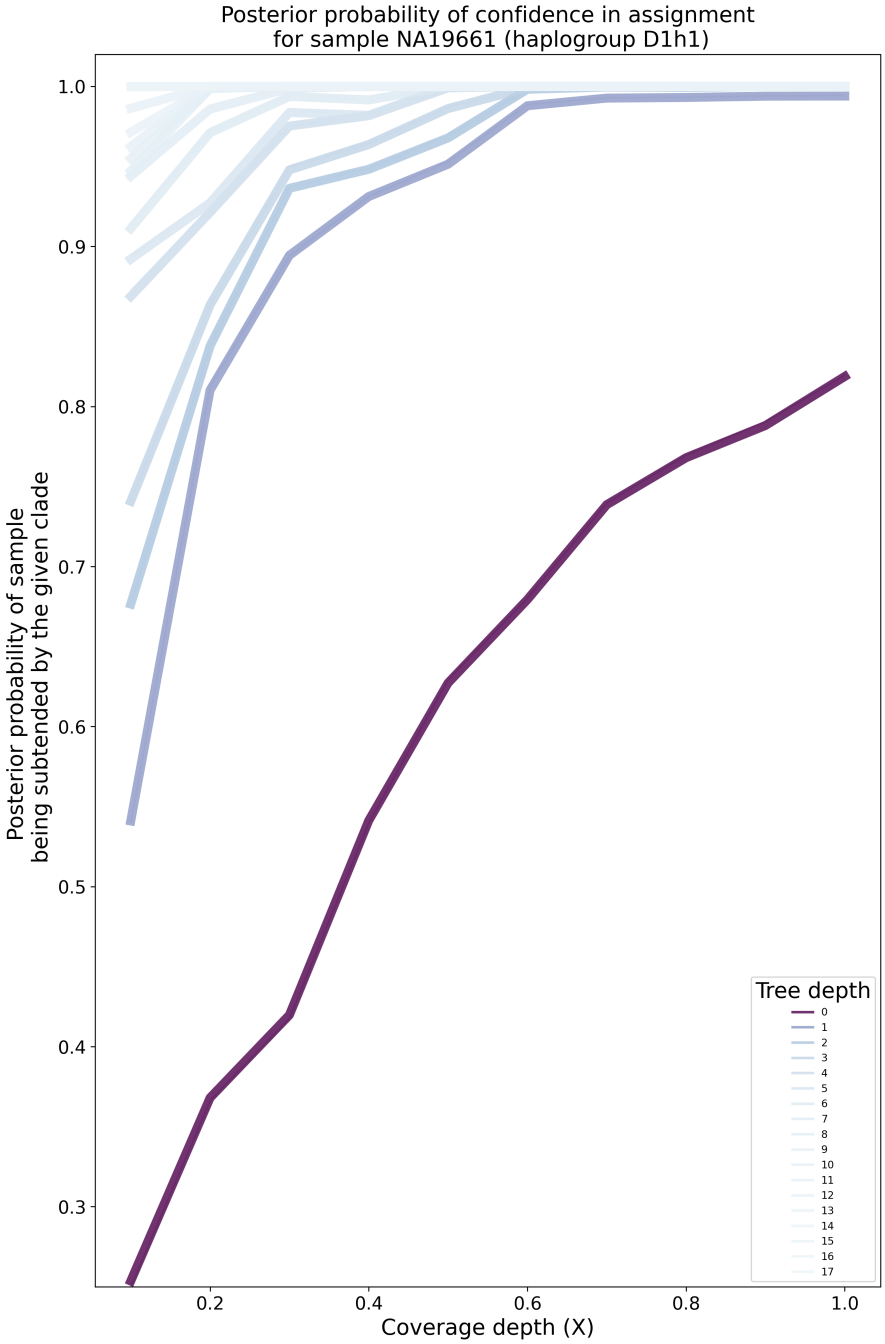

Figure 5: Posterior probabilities of clade-level haplogroup assignment for the Thousand Genomes Project sample NA19661 by target coverage depth (mean over 100 replicates). The darker the lineplot, the shallower (i.e. more recent) the depth of the tree. At a fixed tree depth, the posterior probabilities tend to increase as coverage depth increases. The posteriors asymptotically approach one as the considered clades become more ancestral to the putative haplogroup. The rate of increase is greater for greater target coverage depths, as expected.

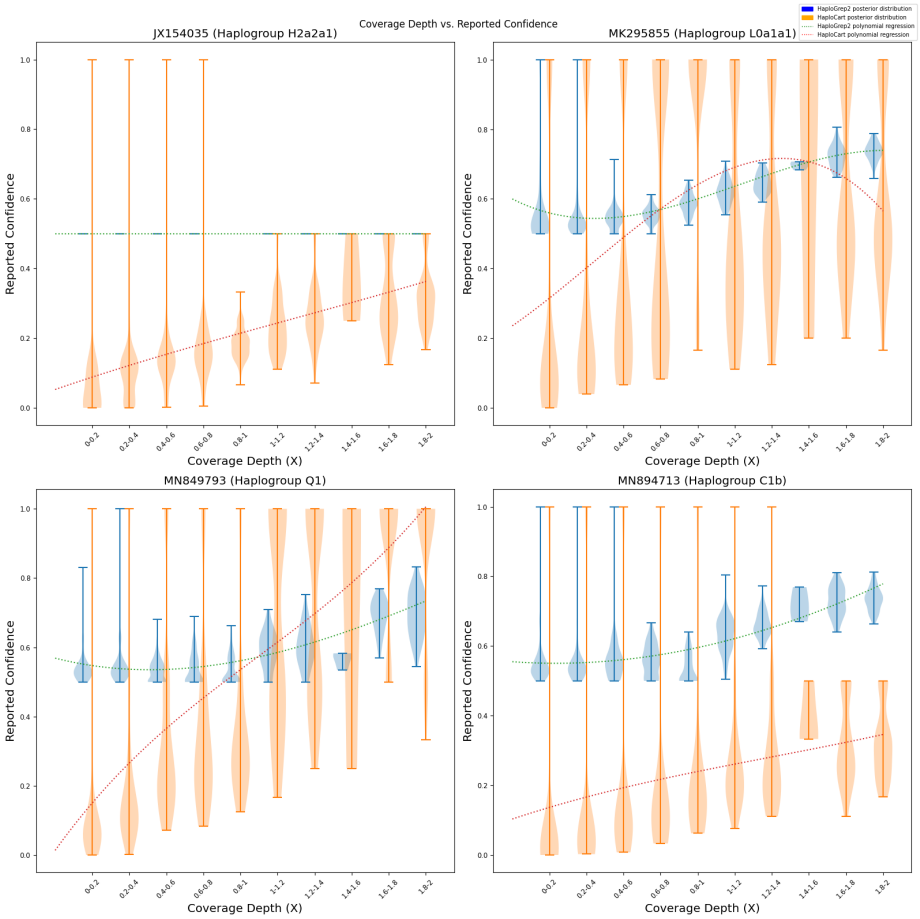

Figure 6: Correlation between Mean Coverage Depth and Reported Confidence Scores on Simulated Paired-end FASTQ Data. Distribution of posterior probabilities for precise haplogroup assignment for reads generated from four different simulated paired-end FASTQ samples (without added NuMT reads). As coverage depth increases, `HaploCart` posterior probabilities tend to increase, which is the desired behavior. In contrast, `HaploGrep2` quality scores obey this behavior only for three of the four samples; the quality score for samples assigned to haplogroup H2a2a1 are always precisely 0.5. `HaploCart` is therefore less biased towards the haplogroup of the sample. Regression curves are polynomials of the third degree.

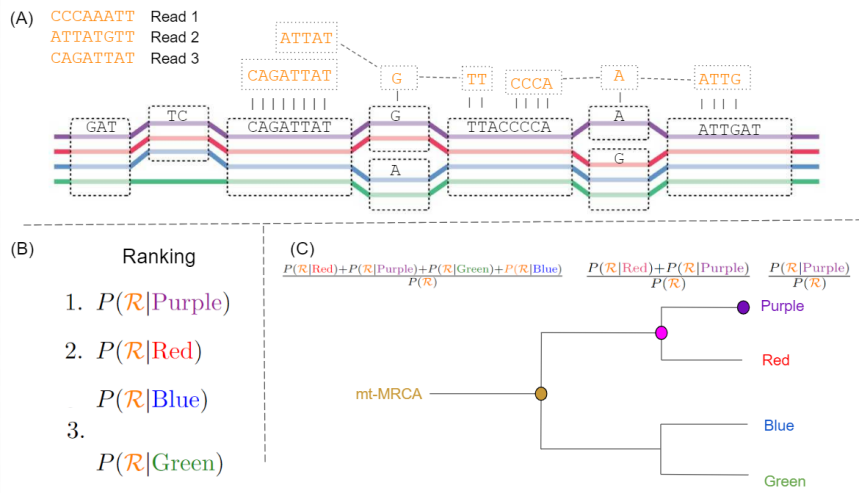

Figure 7: Graphical Representation of the `HaploCart` Inference Algorithm. (A): A variation graph with four embedded haplogroups. Each haplogroup sequence can be reconstructed by walking the appropriate nodes of the graph. Suppose we observe three DNA reads (top left). Read 1 is derived unambiguously from the purple haplogroup. Read 2 is equally likely to have come from the purple or red haplogroup. Read 3 could equiprobably have come from any of the four embedded haplogroups. (B) Based on observation of the reads ($\mathcal{R}$) we compute the posterior probability $P(h_k|\mathcal{R})$ for each embedded haplogroup h_k. In this case the haplogroup which maximizes this quantity is the purple one, which becomes the haplogroup assignment for the sample. (C) HaploCart (optionally) reports the proportion of posterior mass which falls on the assigned haplogroup (purple). It then goes up each ontological level of the tree, up to the mt-MRCA, reporting the proportion of posterior mass for all haplogroups within the relevant clade.

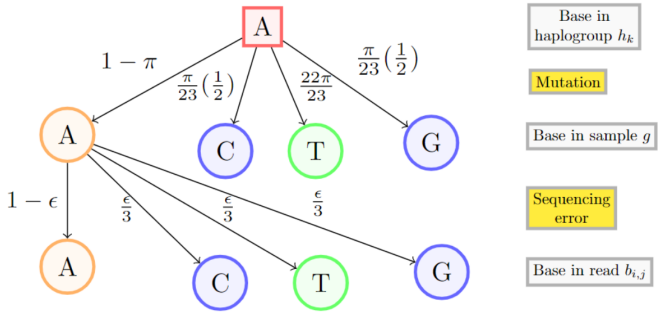

Figure 8: Illustration of $P[b_{i,j}|h_k]$. Probability of observing a given nucleobase under the hypothesis that the sample belongs to a particular haplogroup. The rectangular box is the observed base. The probability of no mutation is $1-\pi$, and the remaining probability mass is distributed with reference to the $\frac{ti}{tv}$ of human mtDNA. Transitions are shown in green, transversions in blue. We assign probability $1-\epsilon$ to the event that no sequencing error has occurred. The remaining probability mass is equidistributed across the other three bases which may be the underlying base in the sample. Not all arrows are shown.

Abbreviations

mt-MRCA: Mitochondrial most recent common ancestor; mtDNA: Mitochondrial DNA; NuMT: Nuclear DNA segment of mitochondrial origin; CLI: Command-line interface; rCRS: Revised Cambridge Reference Sequence