Supplementary Material for "Integer programming framework for pangenome-based genome inference"

S1 Path Inference Problem is NP-hard

Proof for Theorem 1. We begin with an instance $G_H(V_H, E_H)$ of the Hamiltonian Path Problem. Let $V_H = \{u_1, \ldots, u_n\}$. We first create a graph $G' = (V', E')$ where

$$V' = \{s\} \cup \{u_k^i \mid 1 \leq k \leq n, 1 \leq i \leq n\} \cup \{t\}$$

$$E' = \{(s, u_k^1) \mid 1 \leq k \leq n\} \ \cup \ \{(u_k^i, u_h^{i+1}) \mid (u_k, u_h) \in E_H, 1 \leq i < n\} \ \cup \ \{(u_k^n, t) \mid 1 \leq k \leq n\}$$

For $1 \leq x \leq n + 2(c(n+1)+1)$, let $\text{bin}(x)$ be standard binary encoding of x using $b = \lceil \log_2(n + 2(c(n+1)+1)) \rceil + 1$ bits. We assign the vertex labels

$$\begin{aligned}
\sigma(u_k^i) &= \text{bin}(k) \circ 0^b 1 \qquad \text{for } 1 \leq i \leq n, 1 \leq k \leq n \\
\sigma(s) &= \text{bin}(n+1) \circ 0^b 1 \circ \text{bin}(n+2) \circ 0^b 1 \circ \cdots \circ \text{bin}(n + c(n+1) + 1) \circ 0^b 1 \\
\sigma(t) &= \text{bin}(n + c(n+1) + 1 + 1) \circ 0^b 1 \circ \text{bin}(n + c(n+1) + 1 + 2) \circ 0^b 1 \circ \text{bin}(n + 2(c(n+1)+1)) \circ 0^b 1.
\end{aligned}$$

We create a distinct haplotype path for each edge that supports only that edge. We define the set of strings $\mathcal{S} = \{\text{bin}(1) \circ 0^b 1, \text{bin}(2) \circ 0^b 1, \ldots, \text{bin}(n + 2(c(n+1)+1)) \circ 0^b 1\}$. See Figure S1 in Appendix for a small worked example. The reduction presented above clearly runs in polynomial time for $c = |V|^{\Theta(1)}$. Combined with Lemmas 3 and 4, Theorem 1 follows.

Lemma 3. *If G_H contains a Hamiltonian path, then G' has an inferred path $\mathcal{P}$ with $Cost(\mathcal{P}) = c \cdot (n+1)$.*

Proof. Let $u_{i_1}, \ldots, u_{i_n}$ be a Hamiltonian path in G_H. We take as our inferred path $\mathcal{P} = s, u_{i_1}^1, u_{i_2}^2, \ldots, u_{i_n}^n, t$. As every edge has its own corresponding haplotype, the number of recombinations is $n+1$. Furthermore, since $u_{i_1}, \ldots, u_{i_n}$ is a Hamiltonian path and s and t are included in the inferred path, all strings in $\mathcal{S}$ occur in $\sigma(\mathcal{P})$. Hence, the total cost is $c \cdot (n+1)$. □

Lemma 4. *If G' has an inferred path $\mathcal{P}$ with $Cost(\mathcal{P}) \leq c \cdot (n+1)$, then G_H has a Hamiltonian path.*

Proof. First, we claim that s and t must be included in $\mathcal{P}$. The $0^b 1$ substrings are used as padding to prevent any string in $\mathcal{S}$ from being matched using portions of two or more vertex labels. Therefore, if s or t are not

included in the inferred path, at least $c \cdot (n+1) + 1$ strings from $\mathcal{S}$ do not occur in $\sigma(\mathcal{P})$, contradicting that $Cost(\mathcal{P}) \leq c \cdot (n+1)$. Hence, the inferred path $\mathcal{P}$ must contain s and t and be of the form s, $u^1_{i_1}$, …, $u^n_{i_n}$, t for some i_1, …, i_n. Since each edge traversed corresponds to a recombination, the total number of recombinations is $n+1$. The only way the $Cost(\mathcal{P}) \leq c \cdot (n+1)$ is if all strings in $\mathcal{S}$ occur as substrings in $\sigma(\mathcal{P})$. Again, due to the 0^b1 padding in the vertex labels, this can only happen if for all $i \in [1, n]$, u^k_i is a vertex in $\mathcal{P}$ for some k. Furthermore, because there are n vertices in $\mathcal{P}$ that are not s or t, there must be exactly one such k for a given i. We conclude that u_{i_1}, …, u_{i_n} is a Hamiltonian path in G_H. □

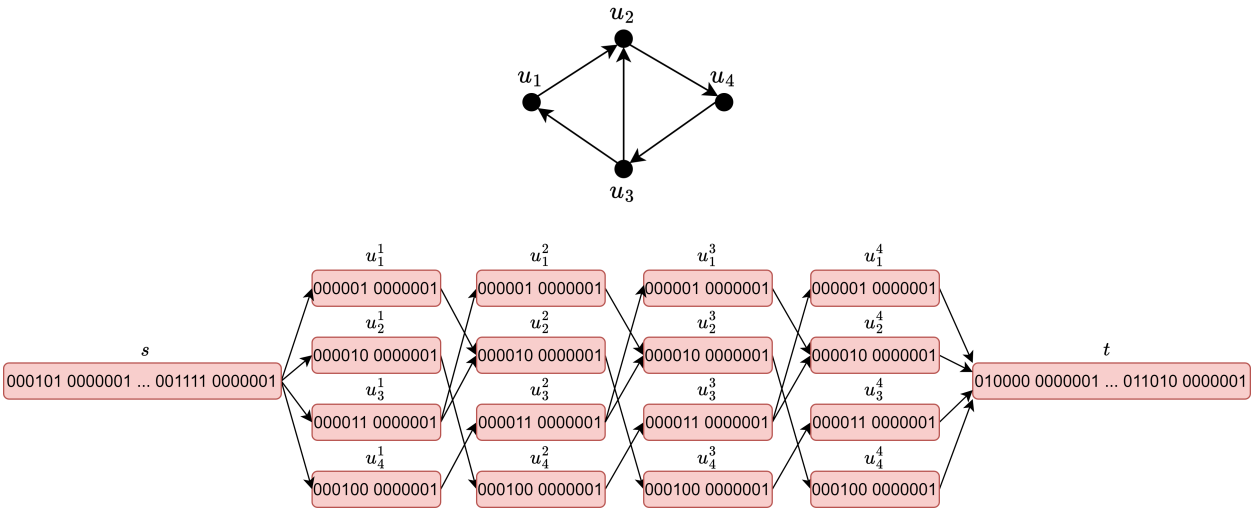

Fig. S1: A small example of our reduction from Hamiltonian Path Problem to Problem 1 (Theorem 1). (Top) The starting instance of G of Hamiltonian Path Problem. (Bottom) The vertex labeled graph G' constructed from G. Here, $n = 4$ and we assume $c = 2$, making $b = \lceil \log_2(n + 2(c(n+1)+1)) \rceil + 1 = 6$. Each edge is supported by a unique haplotype (not shown). The string set is $\mathcal{S} = \{0000010000001, 0000100000001, \ldots, 0110100000001\}$.

Table S1: Additional information about the MHC sequences of five haplotypes (APD, DBB, MANN, QBL, SSTO). We show the length of the complete assembly in the second column. The third and forth columns show edit distance statistics between the assembly and 49 reference haplotypes included in the pangenome reference. In the last two columns, we list the SRA accession numbers and coverage of short-read sequencing datasets.

		Edit distance with pangenome reference haplotypes		Short-read data	
Haplotype	Assembly length (Mbp)	Mean	Minimum	SRA Accession	Coverage
APD	4.93	146,423	37,102	SRR17272303	16.26x
DBB	5.05	174,619	10,380	SRR17272302	12.91x
MANN	5.03	189,464	58,168	SRR17272301	18.20x
QBL	4.90	159,968	72,293	SRR17272300	12.85x
SSTO	5.05	161,044	35,583	SRR17272299	15.04x

Table S2: Commands used for running various tools

Haplotype/Genotype Imputation	
PHI	1) `vcf2gfa.py -v multi-allelic_phased.vcf -r reference.fa > graph.gfa` 2) `PHI -t32 -g graph.gfa -r reads.fq -o imputed_hap.fa`
PanGenie	`PanGenie -t32 -i reads.fq -r reference.fa -v multi-allelic_phased.vcf -o out_vcf_PG`
VG	1) `kmc -t32 -k29 -m128 -okff -hp reads.fq sample tmp_dir` 2) `vg haplotypes -t32 -v2 --num-haplotypes 1 -i input.hapl -k sample.kff -g sample_graph.gbz input_graph.gbz` 3) `vg paths -x sample_graph.gbz -F -S recombination > imputed_hap.fa`
VCF Operations	
Transform VCF to have non-overlapping variants	`vcfbub -l 0 -r 100000 -i input.vcf > output.vcf`
Filter heterozygous variants	`bcftools view -i 'GT="hom"' input.vcf.gz > output.vcf`
Generate haplotype from reference genome and VCF file	`bcftools consensus -f reference.fa -o imputed_hap.fa input.vcf.gz`
Evaluation	
Edit distance	`edlib-aligner ground-truth_hap.fa imputed_hap.fa`

Table S3: Count of homozygous and heterozygous genotype calls made by PanGenie. In our benchmark, we excluded the heterozygous calls because the sequencing datasets were derived from homozygous cell lines.

Coverage	APD		DBB		MANN		QBL		SSTO	
	Hom	Het	Hom	Het	Hom	Het	Hom	Het	Hom	Het
0.1x	52,816	6,245	51,435	7,626	52,452	6,609	53,707	5,354	53,893	5,168
0.5x	56,249	2,812	55,845	3,216	56,258	2,803	56,447	2,614	56,064	2,997
1x	57,448	1,613	57,010	2,051	57,064	1,997	57,224	1,837	57,099	1,962
2x	58,201	860	57,948	1,113	58,334	727	58,101	960	58,397	664
5x	58,552	509	58,382	679	58,601	460	58,340	721	58,228	833
10x	58,533	528	58,478	583	58,188	873	58,343	718	58,337	724
Complete data	58,647	414	58,457	604	58,592	469	58,457	604	58,521	540

Table S4: We report additional performance statistics for PHI on all our datasets. We specify the number of recombinations used in the solution in the second column. Next, we mention the runtime and memory usage of PHI. In the fifth and the sixth columns, we specify edit distance and alignment identity between the output MHC sequence and the ground-truth sequence. Alignment identify is defined as the ratio of the number of character matches divided by the length of the alignment. In the last three columns, we give statistics about the minimizers computed from sequencing reads. We give the count of distinct minimizers observed in the read set. A fraction of minimizers would be absent from the graph, and some fraction would be present in all reference haplotypes, making them 'uninformative'. The matches of only the remaining fraction minimizers are useful while solving the optimization problem.

Coverage	Recombinations	Time (s)	Memory (GB)	Edit distance	Alignment identity (%)	Minimizers (Reads)	Minimizers % Absent | % Uninformative
Haplotype: APD							
0.1×	3	1840	72	7551	99.85	33248	36.33 | 43.12
0.5×	7	1294	84	2272	99.95	156209	37.90 | 41.42
1×	7	2338	93	2220	99.95	289795	41.46 | 39.19
2×	9	2702	108	1948	99.96	508720	46.47 | 35.84
5×	10	4671	125	1779	99.96	984355	59.39 | 27.05
10×	10	3683	134	1810	99.96	1599325	72.22 | 18.33
16.26×	10	4536	134	1810	99.96	2288126	80.17 | 13.00
Haplotype: DBB							
0.1×	2	1604	70	2191	99.96	33901	37.28 | 41.78
0.5×	4	1467	83	1415	99.97	157510	39.66 | 39.60
1×	4	2022	92	1496	99.97	293996	42.54 | 37.84
2×	4	2502	108	1472	99.97	518085	47.59 | 34.28
5×	4	4175	126	1385	99.97	1015730	60.37 | 25.75
10×	4	4525	132	1377	99.97	1660305	72.79 | 17.55
12.91×	4	4743	135	1377	99.97	2028107	77.31 | 14.58
Haplotype: MANN							
0.1×	3	1680	67	41028	99.19	33614	34.31 | 43.07
0.5×	7	1658	85	38379	99.24	153933	36.66 | 41.50
1×	8	2183	94	37898	99.25	288713	39.33 | 39.76
2×	9	3054	109	37728	99.25	502336	44.89 | 36.22
5×	12	3774	126	36263	99.28	964364	57.71 | 27.55
10×	14	5426	132	35941	99.29	1553694	70.85 | 18.86
18.20×	14	4843	134	35940	99.29	2450244	81.06 | 12.15
Haplotype: QBL							
0.1×	3	2222	88	15062	99.69	32464	35.13 | 43.05
0.5×	9	1236	81	7829	99.84	153818	37.47 | 41.77
1×	10	2388	92	4610	99.91	284587	39.92 | 40.35
2×	14	2981	109	3561	99.93	502087	46.98 | 36.14
5×	17	3986	123	3349	99.93	966151	58.80 | 27.40
10×	17	4049	129	3356	99.93	1566636	71.76 | 18.63
12.85×	17	4113	131	3343	99.93	1862566	75.90 | 15.84
Haplotype: SSTO							
0.1×	2	2013	72	17626	99.65	33792	36.06 | 41.98
0.5×	12	1812	84	10471	99.79	156473	37.60 | 41.12
1×	20	2536	93	5150	99.90	291484	41.05 | 38.59
2×	24	2977	108	4671	99.91	513683	46.50 | 35.02
5×	24	5023	124	4611	99.91	992511	59.01 | 26.68
10×	24	5021	132	4634	99.91	1609715	71.88 | 18.16
15.04×	24	4499	137	4637	99.91	2206289	79.07 | 13.44

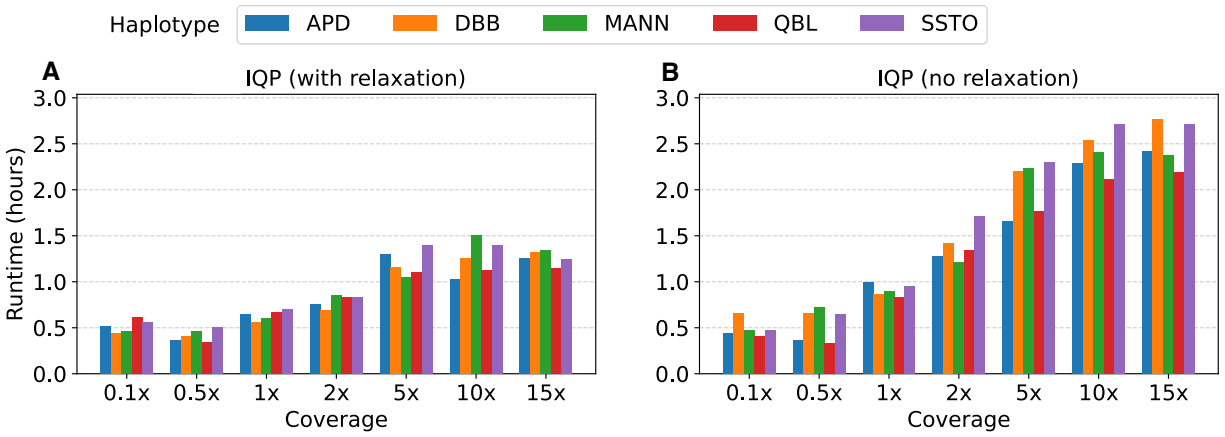

Fig. S2: Evaluation of the performance of the IQP method with and without relaxation of the binary edge variables x_{uv}. We compared runtime using various short-read datasets.

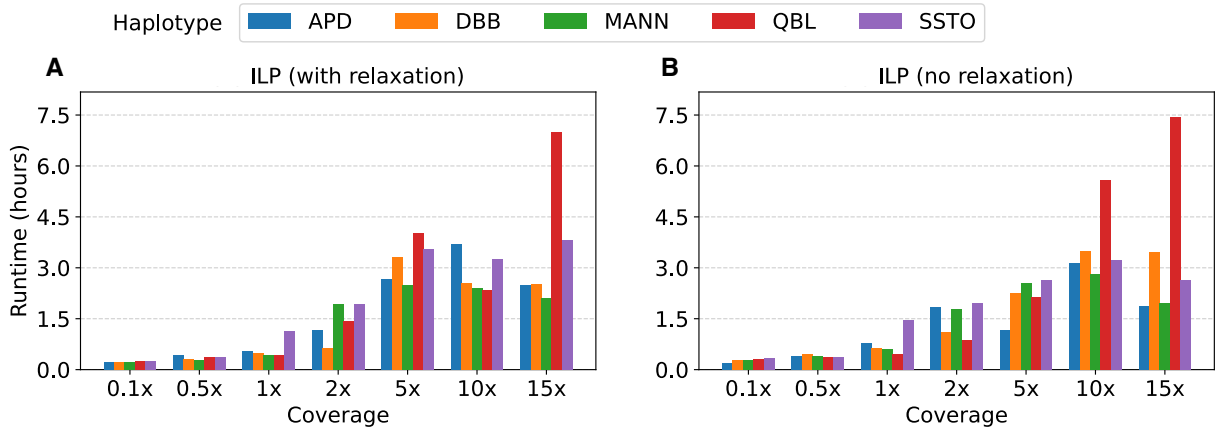

Fig. S3: Evaluation of the performance of the ILP method with and without relaxation of the binary edge variables x_{uv}. We compared runtime using various short-read datasets.