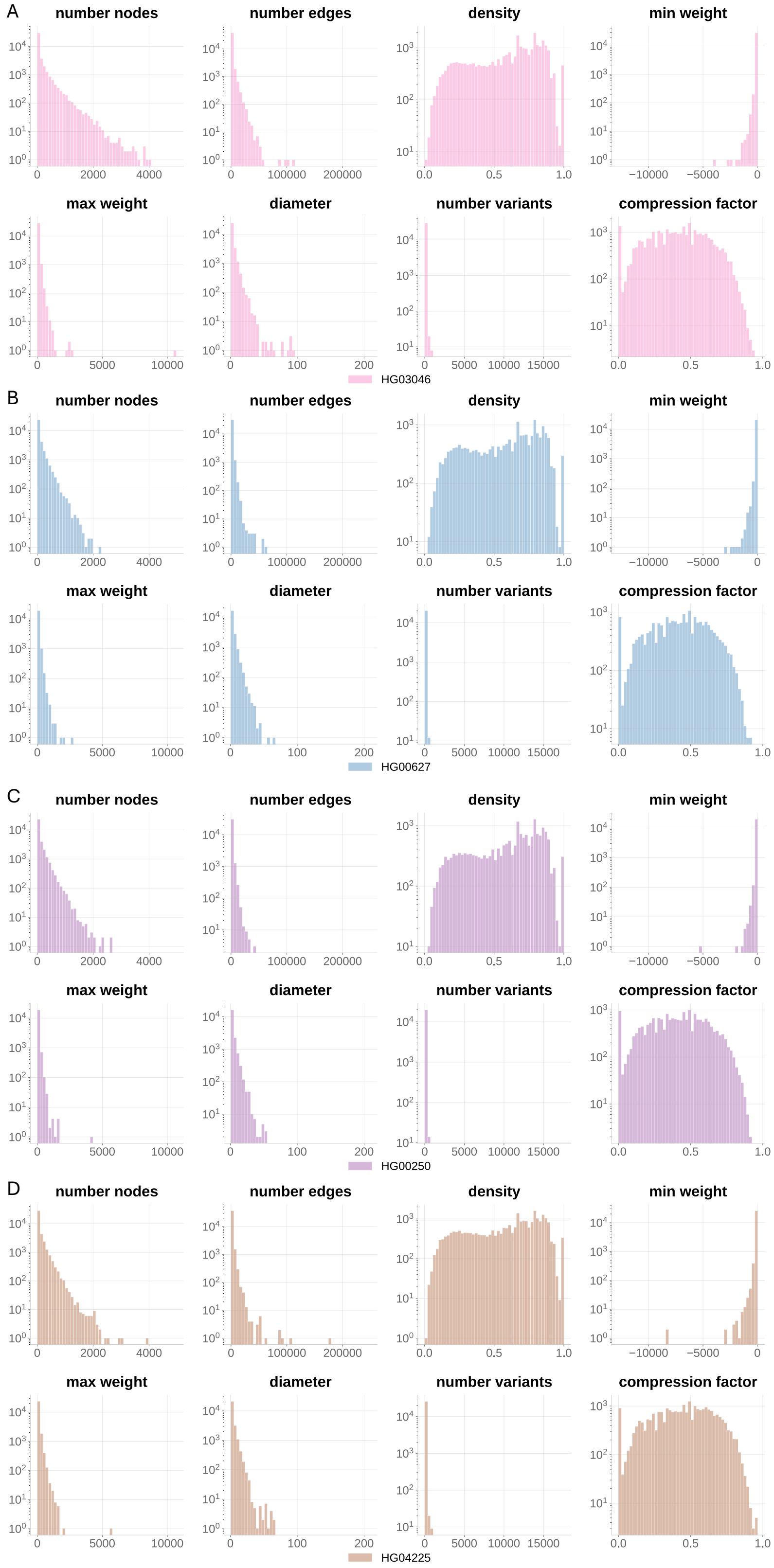
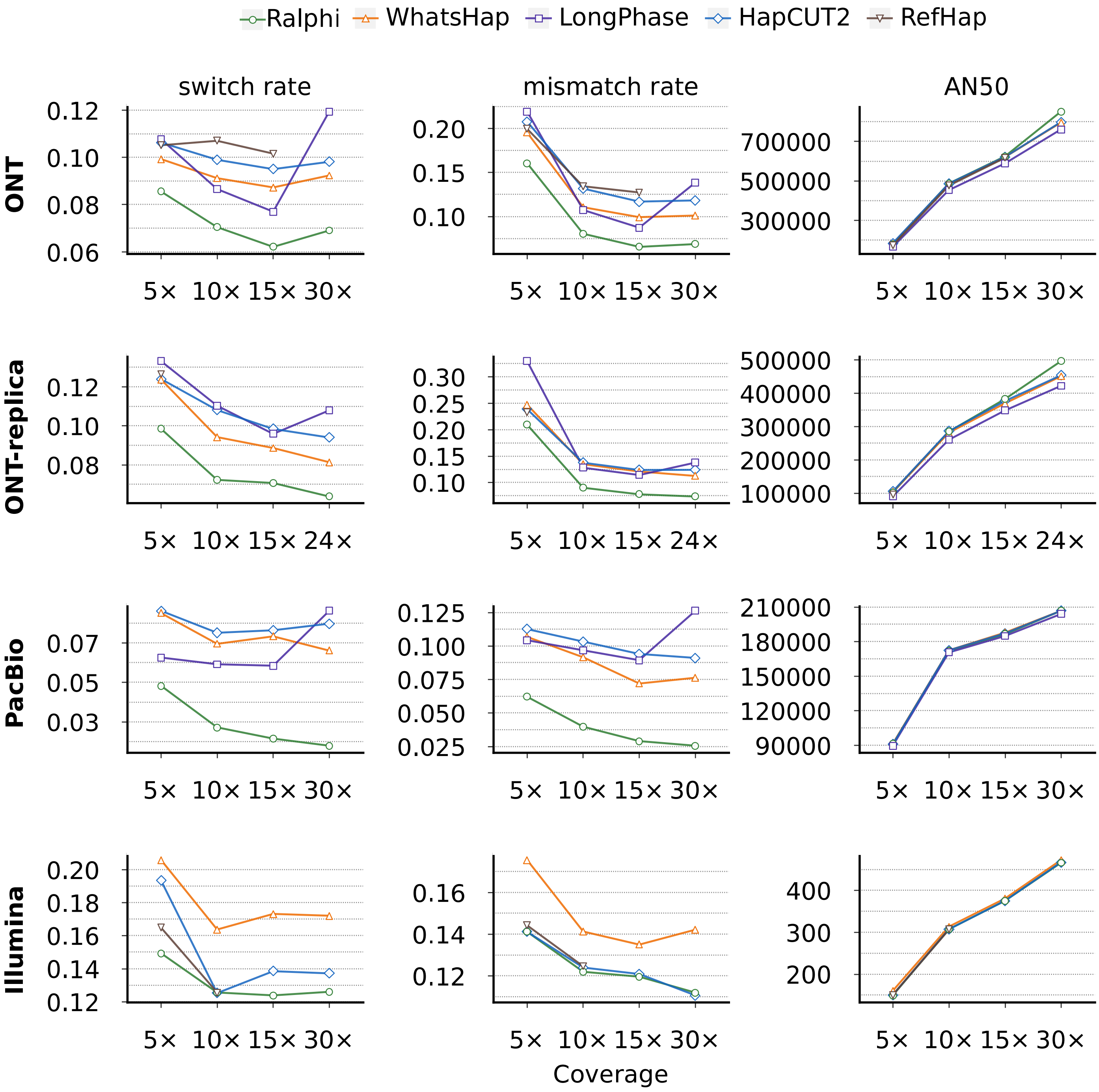
Contents

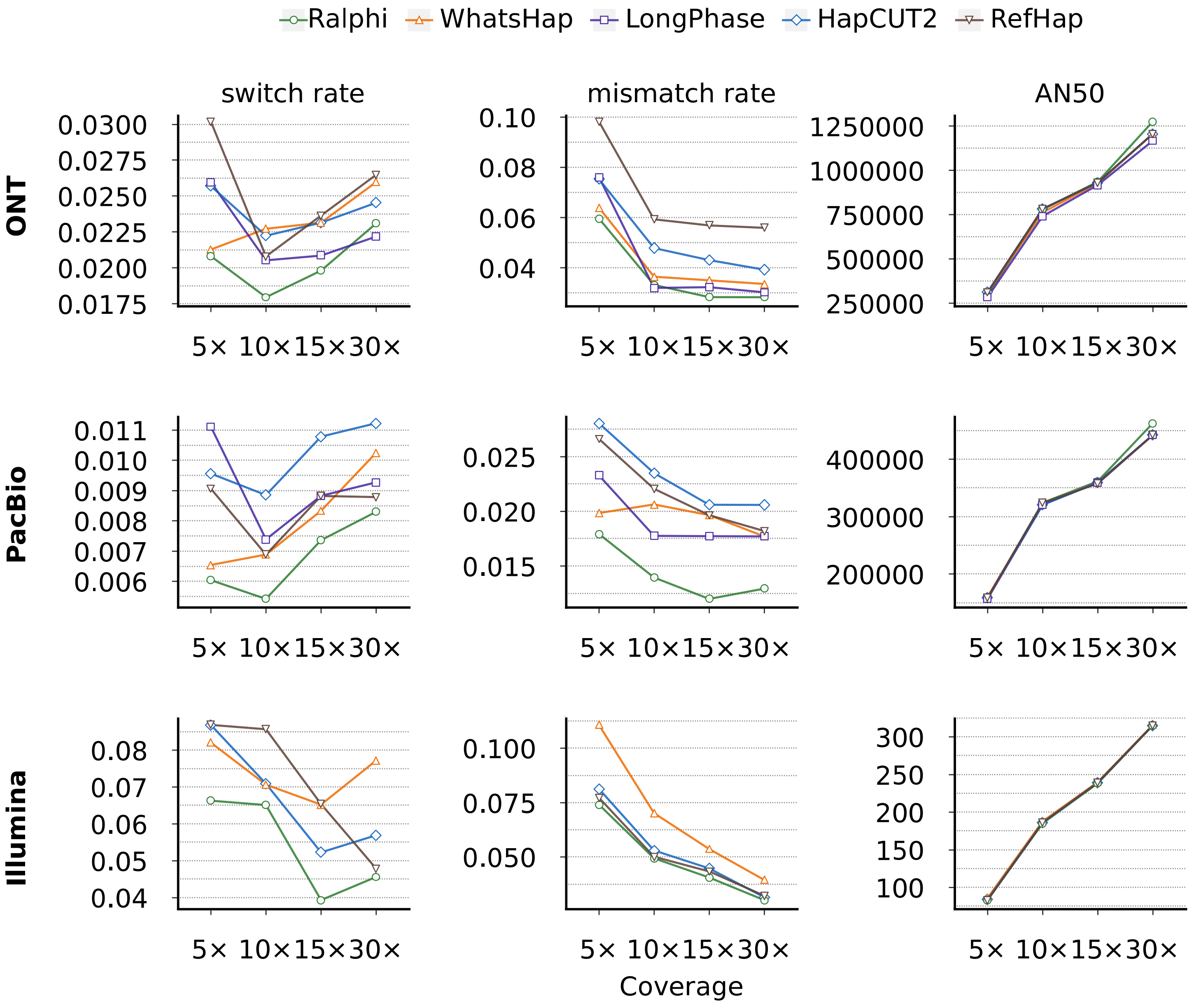
Supplemental Figures



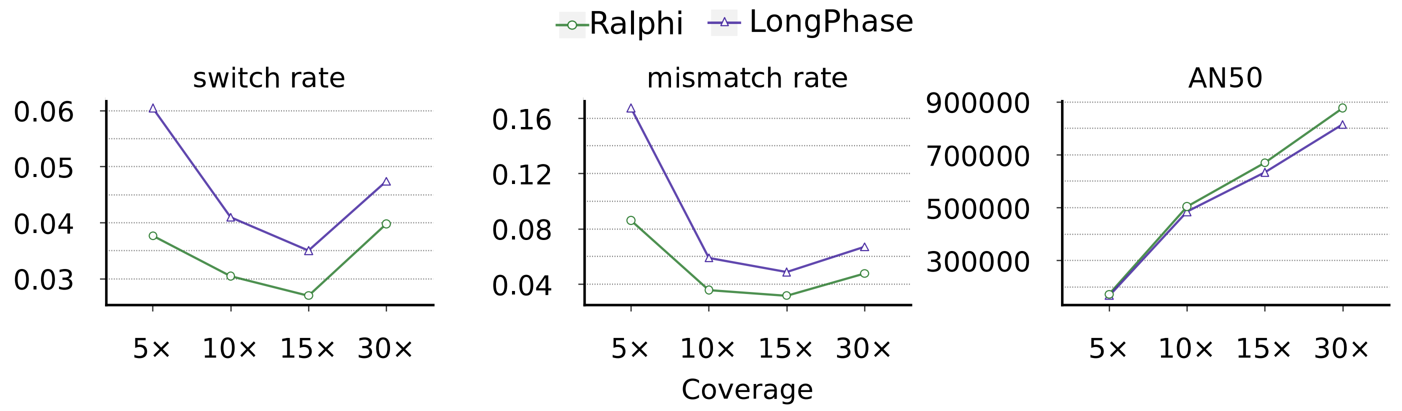
Supplemental Fig. S1: Distributions of fragment graph properties for a sample subset of genomes selected for training (computed over a combination of real and simulated short reads and synthetic ONT reads). **A**. HG03046: African (Gambian). **B**. HG00627: East Asian (Han Chinese). **C**. HG00250: European (British). **D**. HG04225: South Asian (Indian Telugu).



Supplemental Fig. S2: Performance evaluation on chromosome 1 and 20 of NA12878.



Supplemental Fig. S3: Performance evaluation on chromosome 1 and 20 of HG002.



Supplemental Fig. S4: Performance evaluation of Ralphi and LongPhase on NA12878 ONT reads. LongPhase is executed in default mode, while Ralphi is executed without variant filtering on the variants phased by LongPhase.

Supplemental Notes

1 Training data generation

1.1 Synthetic data

Genome simulation

Each synthetic genome `G` was simulated based on the GRCh38 reference and its corresponding variants from the 1000 Genomes Project VCF (`1KG.vcf`) as follows. First we extracted bi-allelic SNPs from `1KG.vcf` using the command: `bcftools view -s <G> <1KG.vcf> | bcftools norm -m + | bcftools view -types snps -I -a -ghet -c1 -Ov -m2 -M2`. We then inserted these variants into the GRCh38 reference using the commands: `bcftools consensus -H 1 -f <REF> -o <G>.hap1.fa` and `bcftools consensus -H 2 -f <REF> -o <G>.hap2.fa` to generate each genome haplotype separately (providing the filtered vcf file as input).

Read simulation

Illumina. We used `dwgsim` to simulate paired-end Illumina reads at 30× coverage from each synthetic genome with varying error rates. Reads from each haplotype were simulated using the command: `-C 15 -1 151 -2 151 -e <err_rate> -E <err_rate> -z 88 -y 0 -S 0 -c 0 -r 0 -R 0 -H` (note: the random seed parameter `-z` was set for reproducibility). The simulated reads were then aligned using `bwa mem` (default parameters) and sorted using `samtools sort`.

ONT. We used `pbsim3` to simulate ONT reads at 30× coverage from each synthetic genome. Reads from each haplotype were simulated using the command: `pbsim -strategy wgs -method qshmm -qshmm data/QSHMM-ON` `-depth 15 -accuracy-mean 0.98 -seed 123` (note: the random seed parameter `-seed` was set for reproducibility). The simulated reads were then aligned using `minimap2` (using the `-map-ont` preset) and sorted using `samtools sort`.

Downsampling. To generate multiple coverage regimes, we downsampled the simulated 30× BAM files using `samtools view` with the subsampling parameter `-subsample` and the seed parameter `-subsample-seed` for reproducibility, these parameters were set as follows: (1) 15×: `-subsample 0.50 -subsample-seed 2` from the 30× BAM, (2) 10×: `-subsample 0.66 -subsample-seed 3` from the 15× BAM, (3) `-subsample 0.50 -subsample-seed 4` from the 10× BAM.

1.2 Real data

All real ONT datasets were first downsampled to 30× from their original coverage using `samtools view` and a fixed random seed: `-subsample-seed 1`. The downsampling ratios were set as follows: (1) HG00142: `-subsample 0.758`, (2) HG00263: `-subsample 0.691`, (3) HG00277: `-subsample 0.723`, (4) HG00326: `-subsample 0.640`, (5) HG00372: `-subsample 0.754`, and (6) HG00463: `-subsample 0.712`. Short-read datasets were downloaded at the 30× coverage. The downsampling from 30× to all other coverages was performed as described in the previous section.

1.3 Fragment graph selection

Fragment graph selection followed a stratified sampling approach to ensure balanced representation across sequencing technologies and graph complexities. Graphs were initially partitioned by sequencing technology (ONT or Illumina reads) and data source (real or simulated reads). Within each partition, graphs were categorized by node count into six bins: 3-9, 10-49, 50-99, 100-199, 200-499, and 500-5000 nodes. Each node category was further subdivided into quartiles based on edge count to capture topological diversity. To address the inherent correlation between graph size and maximum achievable reward, we adjusted the number of selected graphs per node category proportionally to the median edge count of that category, thereby balancing the influence of different graph complexities on the overall reward distribution. For Illumina datasets, all the graphs in the 500-5000 nodes bin were retained, with other categories balanced as described above. For ONT data, which exhibited more limited graph availability, all graphs up to 200 nodes were included, with the two largest categories balanced as described above. An equal number of graphs was selected from real and simulated read sources within each technology. Ten percent of graphs from each category were reserved for validation.

2 Model Training parameters

Model training was performed using the following default parameters: discount factor of 0.98, learning rate of $3e^{-6}$, and validation interval of 30,000 graphs.

3 Benchmarking parameters

Ralphi. For all input datasets, Ralphi was executed using its default parameters set for each sequencing platform.

HapCUT2. To extract fragments, the `extractHAIRS -bam <BAM> -VCF <VCF>` command was executed with default parameters; the parameter `-ont 1` was additionally set for ONT reads and `-pacbio 1` was set for PacBio HiFi reads, respectively; the `-ref <REF>` parameter was set when running on ONT and PacBio long reads. To run phasing, the `HAPCUT2` command was executed using default parameters.

WhatsHap. WhatsHap was executed using the `whatshap phase -reference <REF> -ignore-read-groups -only-snvs <VCF> <BAM>` command; the recommended `-reference` parameter was additionally set for ONT and PacBio inputs; we ran short reads with `-no-reference` since it resulted in better performance.

LongPhase. Longphase was executed with the command `longphase phase -s <VCF> -b <BAM> -r <REF>` and the flags `-ont` for ONT inputs and `-pb` for PacBio HiFi inputs, respectively.

RefHap. The `extractHAIRS` command from HapCUT2 was used for fragment extraction with default parameters, followed by `java -cp SIH.jar mpg.molgen.sih.main.SIH` command to run RefHap phasing.

4 Software versions

The following versions of software tools were used in our benchmarks and for training dataset generation: minimap2 2.28, bwa-mem 0.7.17-r1188, samtools 1.21, bcftools 1.13, htslib 1.21, dwgsim 1.1.14, pbsim3 3.0.5, WhatsHap (2.3 for ONT and PacBio; 2.0 for Illumina since it performed better with this older version on this data type), HapCUT2 (1.3.4), LongPhase (1.5), RefHap (1.0.0).