# Telomere-to-telomere phased genome assembly using error-corrected Simplex nanopore reads

Dominik Stanojević[1,2], Dehui Lin[1], Paola Florez de Sessions[3], Mile Šikić[1,2]

[1]Laboratory of AI in Genomics, Genome Institute of Singapore, A*STAR, Singapore, Singapore
[2]Laboratory for Bioinformatics and Computational Biology, Faculty of Electrical Engineering and Computing, University of Zagreb, Zagreb, Croatia
[3]Oxford Nanopore Technologies, Oxford, United Kingdom

Corresponding Author: Mile Šikić (mile_sikic@gis.a-star.edu.sg)

## Abstract

Telomere-to-telomere phased assemblies are standard expectations. To achieve these for diploid and even polyploid genomes, the contemporary approach involves at least two long-read sequencing technologies: high-accuracy HiFi or Duplex nanopore long reads and ultra-long noisy nanopore reads. Using two different technologies increases the cost and the required amount of genomic DNA. Here, we show that comparable results are possible using error correction of nanopore Simplex ultra-long reads and then assembling them using existing state-of-the-art de novo assembly methods. We have developed the HERRO model based on deep learning, which corrects Simplex nanopore reads longer than 10kbp and with a quality value higher than 10. Taking into account informative positions that vary between haplotypes or segments in duplications, HERRO achieves an increase of accuracy of up to 100-fold. Combing HERRO with Verkko assembler, we achieve high contiguity on human genomes by reconstructing many chromosomes telomere-to-telomere, including chromosomes X and Y. We show that HERRO generalises well to other species and it supports both R9.4.1. and R10.4.1 nanopore Simplex reads. These results offer an opportunity to decrease the genome sequencing cost and apply corrected reads to more complex genomes with different levels of ploidy or even aneuploidy.

## Introduction

The accuracy and contiguity of genome assemblies have significantly improved due to new long-read sequencing technologies, new *de novo* assembly methods and dedicated work of the scientific community on manual curation. Resolving most chromosomes, telomere-to-telomere (T2T)[1] has become a standard, along with the phasing of diploid genomes. The primary unresolved areas are those containing ribosomal DNAs (rDNAs). While the sequences of these 43-kilobase segments are known, determining their precise number, the variations among them, and their exact arrangement remains challenging.

Achieving telomere-to-telomere phased genomes requires using several different technologies, primarily high-quality long reads such as HiFi reads produced by Pacific Biosciences sequencers or Duplex nanopore reads (shown in [2,3]) produced by Oxford Nanopore Technologies (ONT), and ultra-long nanopore read. In addition, for successful phasing, either trio binning using parental reads[4] or cross-linking technologies such as Hi-C, Omni-C or Pore-C is required.

Using different technologies, especially those for long-read sequencing, increases the cost and requires more complex algorithms and more genomic DNA.

Several pipelines rely solely on error correction of Simplex nanopore or noisy PacBio reads to achieve diploid assemblies. These include Canu[5] + purge_dups[6], FALCON-Unzip[7], Flye[8]+HapDup[9,10], Shasta[11], and PECAT[12]. However, according to the results published in a recent study[12], their performance is inferior compared to strategies which utilise both high-quality reads and ultra-long nanopore reads. For example, on an HG002 R10 UL dataset (Table 2), PECAT, the top performer among them, achieved a maximum NG50 of 91.4/80.2 Mbp. In contrast, Hifiasm[13–15] and Verkko[16], which utilise multiple sequencing technologies, surpassed this with NG50 values exceeding 100 Mbp (ref. [16], Results). They outperformed PECAT in phasing, too, achieving lower values of Hamming error rate.

NextDenovo[17], another recently published *de novo* assembler for noisy long reads, does not phase genomes. Hypo-assembler[18] uses short Illumina reads to correct nanopore reads.

Here, we present HERRO, a framework based on a deep learning model capable of correcting Simplex nanopore regular and ultra-long reads. Combining HERRO with the existing *de novo* assemblers, such as Hifiasm and Verkko for diploid and LJA[19] (La Jolla Assembler) for haploid genomes, we achieve phased genomes with many chromosomes reconstructed T2T. HERRO is optimised for both R9.4.1. and R10.4.1 pores and chemistry.

HERRO achieves up to 100-fold improvement in read accuracy while keeping intact the most important sites, including haploid-specific variation and variations between segments in tandem duplications. To accomplish this goal, we combined the high-quality HG002

genome assembly (https://github.com/marbl/HG002), advanced read mapping and alignment tools, and a newly developed deep learning model that utilises convolutional networks and self-attention architecture.

Trained on just a few less complex chromosomes of the HG002 human genome, HERRO shows generalisation to various organisms.

## Results

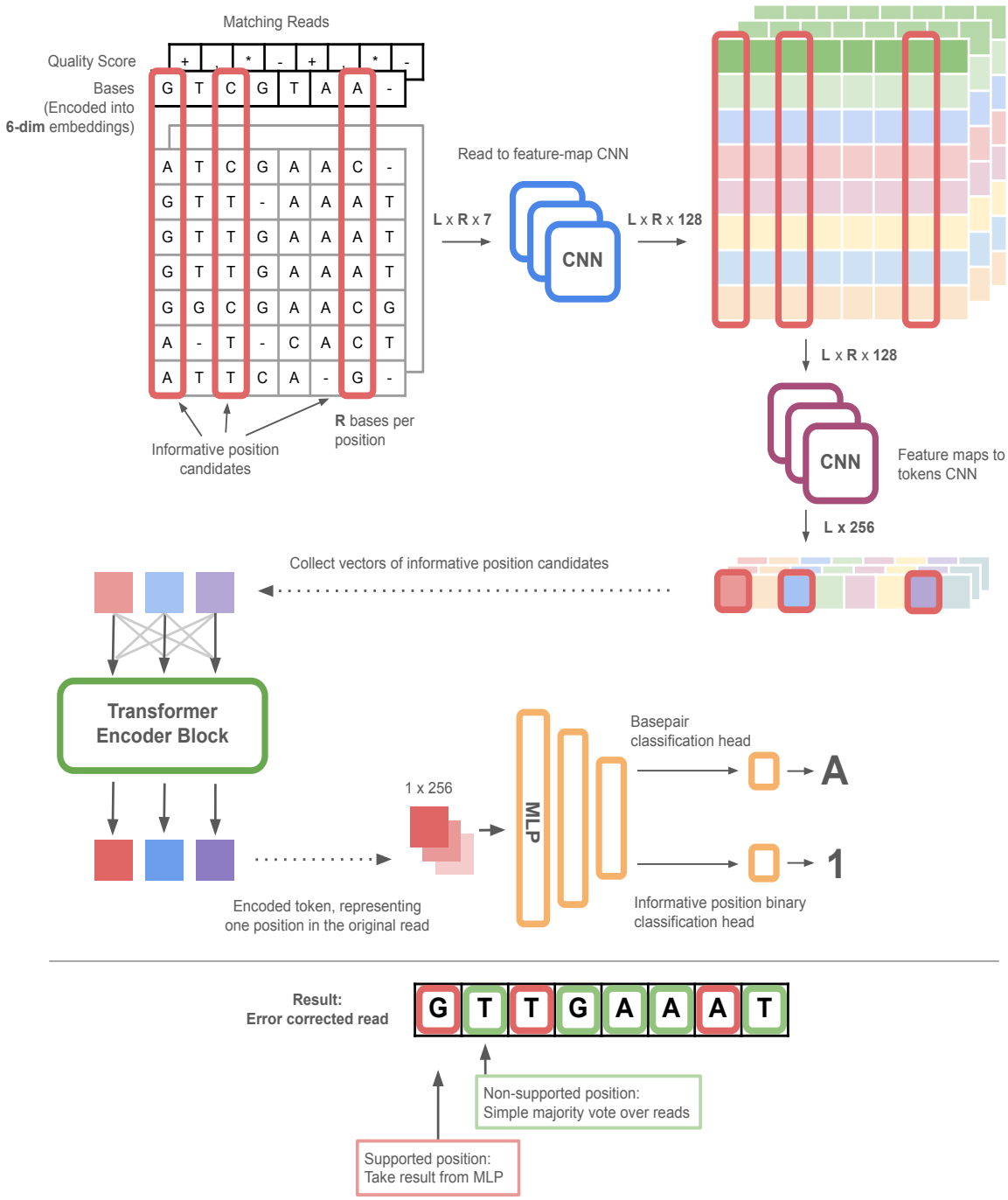



*Fig. 1* ***HERRO deep-learning model:*** *The input to the model consists of embedded bases concatenated with base qualities. The first convolutional block extracts local information for each read independently. The second convolutional block aggregates information from different reads at a specific position in the target read. Next, only supported positions are indexed and fed into the Transformer encoder. The output from the Transformer encoder is used for base prediction and for predicting informative positions for every supported position. Unsupported positions are predicted using majority voting.*

We developed HERRO, a method (Extended Data Fig. 1) aimed at increasing the quality of nanopore reads. HERRO incorporates a newly developed deep learning model (**Error! Reference source not found.**) that combines convolutional blocks with a Transformer encoder. This model processes windows of stacked bases alongside quality scores extracted from all-vs-all pairwise alignments. Initially, the convolutional blocks process these windows to extract local information from each base and to aggregate information across different reads at the same genomic position, generating representations for each position. Subsequently, the model selectively processes supported positions – defined as those having at least two different bases that each appear a minimum of three times - through the Transformer encoder. This encoder is designed to learn interactions between supported positions. By focusing on these challenging positions, HERRO accurately predicts the correct bases and accounts for differences between haplotypes. We define a simple majority voting method for unsupported positions to predict bases. Implementing HERRO significantly reduces error rates and, when combined with top-performing assemblers like Hifiasm and Verkko, it achieves state-of-the-art assembly results, thereby eliminating the reliance on HiFi data.

### HERRO correction leads to highly accurate nanopore reads

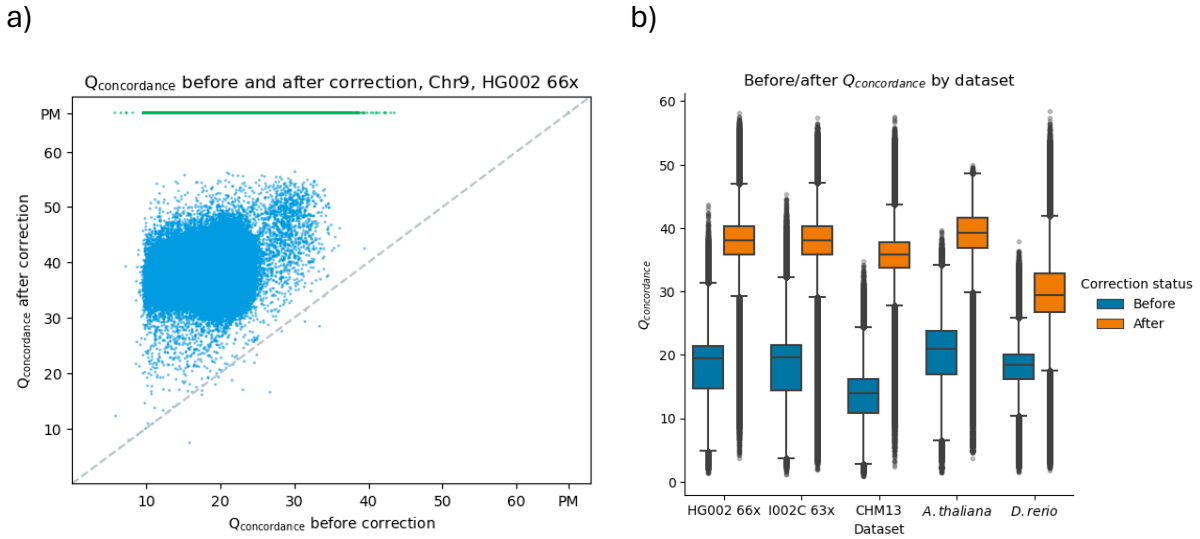



*Fig. 2* ***$Q_{concordance}$ before and after correction****. a) Each point on the plot represents a read aligned to chromosome 9 of the HG002 assembly v1.0.1*[1,16,20–22]*. Points above the dashed diagonal line indicate an improvement in accuracy after correction, while points below it denote a reduction. The "PM" at the end of each axis stands for "Perfect Match" and perfectly matching points are colored green. b) This plot shows the distribution of $Q_{concordance}$ before and after correction across multiple datasets. I002C is a male Indian sample with a recently released reference (Data Availability). For all datasets, alignments that perfectly match the reference were excluded. Chromosomes of HG002 used to train the HERRO model were excluded (1, 2, 17, 18, and 20). The reads from HG002, I002C, and D. rerio were sequenced using the R10.4.1 pore with a 4 kHz sampling frequency, while A. thaliana was sequenced using R10.4.1 with a 5 kHz frequency. CHM13 data were sequenced using R9.4.1. pore. All datasets shown here are ultra-long reads, except for A. thaliana. Whiskers on the plots extend to 1.5 times the interquartile range. The coverage levels, 66x for HG002 and*

*63x for I002C, indicate input coverage for correction in specific runs, as multiple runs were conducted with varying coverages (see Supplementary Tables 2 and 3 for details).*

We evaluated the accuracy of the HERRO corrected reads against a reference for several datasets for which we have high-quality reference assemblies. For human samples, this included HG002 data from Human Pangenome Reference Consortium[23] (HPRC) and Genome In a Bottle[24] (GIAB), CHM13 data from the Telomere-to-Telomere (T2T) consortium[1,20] and I002C data, which is a male Indian sample with a recently released high-quality assembly (https://github.com/lbcb-sci/I002C or https://github.com/LHG-GG/I002C). We also used *D. rerio* (zebrafish, TU strain) data from the Vertebrate Genomes Project[25] (VGP) and *A. thaliana* (thale cress, col-0 strain) data, which has a high-quality assembly[26] (Data Availability).

We aligned both uncorrected and corrected reads to the references and used bamConcordance[27] to obtain a read quality measure ($Q_{concordance}$) and to count different types of errors before and after correction (Methods, Analysis methods). Only primary alignments were considered in the evaluation. For the intersecting set of uncorrected and corrected reads, we observed that the overwhelming majority of reads achieved significant improvements in $Q_{concordance}$ (**Error! Reference source not found.**a). There is a significant upward shift in the overall distribution of $Q_{concordance}$ values for all human and non-human datasets. (**Error! Reference source not found.**b).

| ***Dataset*** | **Correction** | **Mismatch per 10 kbp** | **Non-Hp Ins per 10 kbp** | **Non-Hp Del per 10 kbp** | **Hp-Ins per 10 kbp** | **Hp-Del per 10 kbp** | **Errors per 10 kbp** |
|---|---|---|---|---|---|---|---|
| *HG002* | Before | 133.15 | 45.56 | 58.54 | 26.22 | 71.02 | 334.48 |
| | After (66x) | 0.23 | 0.63 | 0.74 | 0.41 | 1.77 | 3.78 |
| *I002C* | Before | 186.86 | 58.76 | 79.74 | 33.41 | 99.44 | 458.21 |
| | After (63x) | 0.22 | 1.01 | 1.72 | 0.50 | 2.18 | 5.64 |
| *CHM13* | Before | 158.63 | 74.75 | 91.23 | 46.91 | 128.49 | 500.01 |
| | After | 0.38 | 0.34 | 0.85 | 1.23 | 1.60 | 4.39 |
| *A. thaliana* | Before | 38.59 | 27.84 | 22.82 | 10.03 | 29.09 | 128.37 |
| | After | 0.70 | 1.37 | 1.92 | 0.05 | 2.85 | 6.89 |
| *D. rerio* | Before | 71.84 | 29.64 | 50.04 | 17.27 | 41.64 | 210.43 |
| | After | 0.41 | 2.23 | 15.15 | 0.10 | 3.88 | 21.77 |

*Table1 1* ***Counts of errors before and after correction, by type.*** *We used bamConcordance to obtain counts of errors in reads before and after correction, based on alignments to a high-quality reference. In addition to the total number of errors,*

*counts are provided for different categories: mismatch, non-homopolymer insertion, non-homopolymer deletion, homopolymer insertion, and homopolymer deletion.*

Counts of different types of errors in the reads before and after correction were further examined (Table 1). HERRO significantly reduced the error counts for every error type, achieving a more than 50-fold reduction in the total number of errors across multiple datasets. Even for the *D. rerio* dataset, which showed the smallest reduction, the total errors decreased by nearly tenfold. The most substantial decrease was observed in mismatch errors, which dropped from more than 1000 errors per 100 kbp on average in human datasets to fewer than 10. Similar reductions were noted for other types of errors.

**HERRO enables T2T assemblies with nanopore reads as a HiFi replacement**

We evaluated the performance of HERRO-corrected reads in de-novo assembly using several assemblers: Hifiasm, Verkko and LJA (only for haploid datasets). In addition to the datasets used to measure read accuracy, we included human samples HG02818 from HPRC, HG005 from GIAB and a Drosophila melanogaster dataset (Data Availability). For all human datasets except CHM13, we used the trio binning and ultra-long mode settings in Hifiasm and Verkko. HERRO-corrected nanopore reads served as a substitute for highly accurate HiFi reads. Corrected reads were downsampled to ~35x for assembly if their coverages were significantly higher than 35 (38x or above) to control the effect of input coverage on assembly results and reduce computational requirements. Unless stated otherwise, uncorrected and unfiltered (by base quality and length) were used as the ultra-long input to Hifiasm and Verkko (Methods, Data Preprocessing). Hifiasm was configured to use only ultra-long reads longer than 50 kbp. HERRO-corrected reads were chopped to 30 kbp and filtered to retain only those with at least 10 kbp length for Hifiasm (Methods, Analysis methods). Corrected reads were used as they are (without chopping) for Verkko. Since both Hifiasm and Verkko require UL reads, we additionally fed them with uncorrected UL reads, taking care of the minimal required read length. LJA only used the corrected reads as input. For details, see Methods, Analysis methods.

*Diploid human samples*

Multiple diploid human datasets, including HG002 (not the same used for training), I002C (Table 2), and HG02818 (Supplementary Table 1), were assembled. For each dataset, more than half of the 46 chromosomes were obtained as T2T contigs or scaffolds. Verkko generally achieves higher contiguity and produces more T2T chromosomes, whereas Hifiasm assemblies tend to have higher QV for some datasets (Supplementary Tables 1-4).

In multiple runs, the X and Y chromosomes were assembled as T2T contigs or scaffolds, with some runs achieving T2T contigs for both. While Hifiasm successfully assembled the X chromosome as T2T in some runs, only Verkko managed to assemble the challenging[20]

chromosome Y as T2T. Remarkably, using Verkko, we were able to obtain T2T contigs of the X and Y chromosomes even without parental data. These contigs were evaluated against HG002 v1.0.1's X and Y chromosomes using QUAST[28] (Methods, Analysis methods), achieving genome fractions of 100% and 99.98% for X and Y, respectively, with a total of 7 misassemblies (Table 3).

| ***Assembly*** | **Haplotype** | **Size (Gb)** | **NG50 (Mb)** | **NGA50 (Mbp)** | **QV** | **Hamming, switch errors (%)** | **Single-copy genes lost, duplicated, multi-copy genes lost (%)** | **Number of T2T contigs (plus scaffold)** |
|---|---|---|---|---|---|---|---|---|
| *HG002 66x (Verkko trio)* | Maternal | 3.120 | 135 | 134 | 55.86 | 0.06,0.03 | 0.67,0.25, 5.34 | 9(13) |
| | Paternal | 2.999 | 134 | 111 | 54.96 | 0.02,0.02 | 0.85,0.28, 8.03 | 9(12) |
| *HG002 43x HiFi + 31 UL (>100kbp) (Verkko trio)* **Cheng et al 2024**[15] | Maternal | 3.026 | 112 | 110 | 56.50 | 0.07,0.04 | 0.91,0.21,5.29 | 4(5) |
| | Paternal | 2.946 | 134 | 133 | 55.95 | 0.04,0.03 | 0.92,0.30,7.98 | 3(10) |
| *HG002 50x Duplex + 30× UL (Verkko trio)* ***Koren et al 2024***[2] | Maternal | 3.037 | 147 | 114 | 53.59 | 0.05,0.03 | 0.78,0.34,5.19 | 6(11) |
| | Paternal | 2.943 | 134 | 134 | 53.23 | 0.03,0.02 | 0.85,0.26,8.03 | 10(15) |
| *I002C 63x (Verkko trio)* | Maternal | 3.057 | 146 | 124 | 61.06 | 0.29,0.27 | 0.58,0.27,4.34 | 6(13) |
| | Paternal | 2.988 | 134 | 111 | 55.74 | 0.19,0.18 | 0.82,0.31,5.89 | 5(12) |
| *CHM13 (LJA)* | Haploid | 3.052 | 93 | 92 | 47.52 | N.A. | 0.04,0.009,0.55 | 4(4) |
| *D. rerio (LJA)* | Haploid | 1.451 | 56 | 56 | 53.26 | N.A. | 4.79,0.22,22.29 | 7(7) |
| *A. thaliana (Hifiasm)* | Haploid | 0.133 | 26 | 26 | 48.40 | N.A. | 0.05,0.02,0.87 | 3(3) |

*Table 22* ***Metrics of the assemblies for samples with a high-quality reference.*** *We evaluated the assemblies of HERRO-corrected reads using high-quality references. NG50 and NGA50 metrics were calculated with minigraph[29], while quality value (QV), switch errors, and Hamming errors were assessed using Merqury[30]. Gene completeness and duplication levels were measured with asmgene[31] against CHM13 v2.0 for human samples and against high-quality assemblies for non-human samples. T2T contigs and scaffolds were identified using a script that searches for canonical telomeric motifs on the sides of assembled sequences and their alignment to a reference (Methods, Analysis methods). Sequences shorter than 500 kbp were excluded from the assembly analysis for A. thaliana (Supplementary Table 6). For comparison on the HG002 genome when HiFi + UL reads are used, we added results evaluated from assembly for Cheng et al. 's recent publication[15]. They used an UL dataset, extracting only reads longer than 100kbp. For comparison on the HG002 genome when Duplex + UL reads are used, we added results evaluated from assembly from Koren et al.'s recent preprint[2]. All assembly evaluations were done on scaffolds whenever scaffolds were produced.*

We ran experiments at various coverage levels for each diploid ultra-long human dataset to evaluate the impact of different data coverages on correction and assembly. Most of these runs were performed on I002C data, which was not used for training and for which a high-quality reference is available. We observed a consistent increase in contiguity and the

number of T2T chromosomes obtained as the data coverage increased. For instance, with 88x of input data and reads corrected at 71x, we achieved an NG50 of 146 Mbp for both haplotypes of I002C. At lower coverages, around 45x of input data with reads corrected at approximately 35x, the NG50 ranged from about 90 Mbp to 130 Mbp (Supplementary Tables 1-3).

To isolate the effect of correction input coverage on the quality of reads and, consequently, on the assembly, we used I002C reads corrected at 63x and 71x as high-quality inputs. These were used alongside the same uncorrected UL reads, fixed at 45x coverage as the UL input for Verkko (Supplementary Table 3). We observed an increasing trend in the number of T2T contigs or scaffolds as the coverage of the input reads to correction increased, with counts rising from 15 to 21 to 28. Due to a crash with the default Verkko version (v2.0) during the 71x run, a newer version of Verkko was used. To assess the effect of the version difference, we reran the assembly for the 45x run using the newer version of Verkko and found no change in the number of T2T chromosomes.

Additionally, we calculated yak[13] QVs for the I002C reads corrected at different input coverages, finding that QVs increased as coverage increased (Supplementary Table 9). In the table, we also added an evaluation of the HG002 trio Verkko assembly from Cheng et al. paper[15], where they combine the HiFi and UL dataset and the HG002 trio Verkko assembly from Koren et al[2]., where they combine the Duplex and UL dataset. The results are comparable, showing the power of correction of nanopore reads.

Finally, to test on non-UL human data, we corrected and assembled the HG005 dataset, which contains only 10x coverage of reads that are 50 kbp or longer. We obtained 17 T2T chromosomes for HG005, two of which were complete contigs without gaps (Supplementary Table 4). For some extra results on experimental high-accuracy ultra-long reads, see Supplementary Table 10. The results show that error correction leads to more contiguous assembly, more T2T-resolved chromosomes and higher quality values.

| ***Assembly*** | **Genome fraction (%)** | **Misassemblies** | **Mismatches per 100kbp** | **Total mismatches** | **Indels per 100kbp** | **Total Indels** | **Indel <=5 bp** | **Indel >5bp** |
|---|---|---|---|---|---|---|---|---|
| *ChrX, trio* | 100 | 2 | 0.19 | 286 | 8.43 | 13018 | 12672 | 346 |
| *ChrY, trio* | 99.98 | 5 | 1.18 | 736 | 4.08 | 2544 | 2410 | 134 |
| *ChrX, no-trio* | 100 | 2 | 0.17 | 270 | 8.43 | 13007 | 12656 | 351 |
| *ChrY, no-trio* | 99.98 | 5 | 1.16 | 724 | 4.07 | 2539 | 2405 | 134 |

*Table. 33* ***Quast evaluation results of assembled T2T contigs of chromosomes X and Y.*** *Assembled T2T contigs of chromosomes X and Y using corrected HG002 reads were evaluated against HG002 v1.0.1 using Quast. The reads were corrected at 66x coverage, assembled with Verkko, with and without using parental data, respectively. All the misassemblies are relocations.*

### *Difference between genomes*

While evaluating error rates and assembly quality, we noticed the discrepancy between results depending on the reference genome we used in the evaluation and decided to investigate in detail.

Table 4 shows aggregated error rates of corrected reads evaluated by mapping to different references. Since the error rate of corrected reads is very low, it can be almost neglected. The results show that differences in mean mismatch and indel rate are 0.15 – 0.18 % and 0.77 – 0.87 %, respectively, giving us a rough estimation of differences between haploids of human genomes. In addition, we see notable differences in the NGA50 measure when we evaluate genome assemblies using different references. The difference could be more than 20 % (Supplementary Table 12). It is important to note that differences are high even when using another haploid of the same genome as a reference.

| ***Corrected reads dataset*** | **Reference** | **Mean error rate (%)** | **Mean mismatch rate (%)** | **Mean indel rate (%)** | **Median error rate (%)** |
|---|---|---|---|---|---|
| *HG002 66x* | HG002 | 0.033 | 0.0027 | 0.030 | 0.014 |
| | CHM13 | 0.92 | 0.15 | 0.77 | 0.17 |
| | CN1 | 1.0 | 0.18 | 0.87 | 0.19 |
| | I002C | 1.0 | 0.17 | 0.85 | 0.18 |
| *I002C 63x* | HG002 | 0.99 | 0.17 | 0.82 | 0.18 |
| | CHM13 | 0.96 | 0.17 | 0.79 | 0.18 |
| | CN1 | 0.97 | 0.17 | 0.80 | 0.18 |
| | I002C | 0.053 | 0.0054 | 0.048 | 0.014 |

*Table 4* ***The aggregated error rates of HERRO-corrected reads with respect to different assemblies.*** *HERRO-corrected reads for the HG002 and I002C datasets were aligned to several high-quality assemblies to estimate the error rates of reads with respect to them. Both corrected read datasets were aligned to CHM13 v2.0 and CN1[32] v0.8 curated. HG002 reads were aligned to the whole HG002 v1.0.1, and I002C v0.4 maternal plus its Y chromosome and mitochondrial genome. I002C reads were aligned to the whole of I002C v0.4 with the mitochondrial genome, the HG002 v1.0.1 paternal haplotype, and the X chromosome and the mitochondrial genome.*

### *Haploid non-human samples*

We corrected and assembled three homozygous non-human samples: *A. thaliana*, *D. melanogaster*, and *D. rerio*, with only *D. rerio* being ultra-long. For *A. thaliana*, 3 out of 5 chromosomes were assembled as T2T (telomere-to-telomere) contigs (Table 2). LJA

significantly outperformed Hifiasm and Verkko in assembling *D. rerio*, achieving 4 T2T contigs compared to zero and one, respectively (Supplementary Table 5). Furthermore, when we used *D. rerio* corrected reads longer than 90 kbp for assembly with LJA, the number of T2T contigs increased from 4 to 7 (Supplementary Table 5).

*R9.4.1 data*

To test the R9.4.1 ultra-long nanopore reads, we corrected and assembled CHM13. We achieved an NG50 of around 90 Mbp and obtained up to 5 T2T chromosomes as contigs (Supplementary Table 8). While the assembly with Hifiasm resulted in the highest number of T2T chromosomes, the LJA assembly demonstrated better NG75 values and more complete genes, as calculated using asmgene.

## Discussion

High-quality HiFi reads opened new opportunities for reconstructing long and complex genomes. Their low error rate enabled the assembly of some complex regions, such as segment duplication, utilising tiny differences between each segment. In addition, they helped in phasing. However, it was important to include ultra-long nanopore reads to reconstruct more complex regions and find a path through the assembly graph. Due to their higher error rate, their usage in graph construction was limited, leading to graphs primarily based on HiFi reads. This, today's dominant strategy with the extraction of parental information using trio binning and manual curation, enabled several breakthroughs, including the reconstruction of human chromosomes X and Y, telomere-to-telomere reconstruction of a haploid human genome and diploid HG002 genome with high accuracy.

We started our work to utilise the full power of ultra-long reads without the need for HiFi or Duplex nanopore reads. To achieve this, we need to error-correct reads. However, a straightforward error correction, like in the Racon paper[33], traversing POA graphs was insufficient because it might delete important, informative positions that would help reconstruct segmental duplication and phasing. Therefore, we focus our error correction on those positions. As mentioned in the introduction, several methods have already tried this approach. However, even after error correction, they could not[12] use a state-of-the-art assembler like Hifiasm or Verkko.

Using the HERRO method, which combines alignment to high-quality reference with a deep learning approach taking care of informative positions, we are able to create reads that can be fed into existing assemblers. However, they were not designed to exploit the power of ultra-long accurate reads fully. Inspired by the RAFT paper [34], we chopped corrected UL reads as input to HiFiasm, improving the results. For Verkko and LJA, that was not necessary. There

were some issues with Verkko, but they were resolved (private correspondence with Sergey Koren).

The results indicate that we can achieve high contiguity comparable to approaches that combine HiFi or Duplex with ultra-long reads. However, we believe that current assembly results are not optimal due to several factors, including the remaining errors in HERRO and the need for assemblers to be tailored for this type of reads. Additionally, we demonstrated that using the R10.4.1 model, we can successfully correct experimental Q28 Simplex reads (https://labs.epi2me.io/gm24385_ncm23_preview/), leading to improved assembly. Training on this data type could yield even better results. Notably, we were able to accurately reconstruct telomere-to-telomere human chromosomes X and Y. This achievement gives us hope that with further improvements in error correction and optimization of assemblers for accurate ultra-long reads, we will be able to assemble and phase all human chromosomes T2T, as well as other genomes.

We know there is space for improvement in all parts of the pipeline, including better alignment (i.e., using optimised pairwise aligners such as Edlib[35] or A*[36]), reduced memory and computational requirements, optimising window length and better understanding of the remaining errors. Currently, HERRO handles indels up to 30 long nucleotides, and exploring options for longer indels is important. Furthermore, increasing datasets add remaining human chromosomes and training models on different organisms would be necessary for higher read accuracy. Assembly of metagenomics samples with high heterozygosity are another possible field to explore [37–39]. Moreover, we didn't explore assemblers tailored for Simplex nanopore reads that produce collapsed haploid assemblies such as Raven[40] and Flye[8]. They do not phase genomes but might provide meaningful assemblies without requiring UL reads, decreasing cost and required genomic material. Finally, using information about potential methylation (i.e., using Rockfish[41] or similar tools) might improve the calling of informative positions.

We would like to highlight the importance of the reference for training and evaluation. From the results, the average differences between available complete human haploids are 0.15-0.18% and 0.77-0.87 %, measured in mismatches and short indels, respectively. We also found that assessments of results vary depending on the reference used for calculating the evaluation measures such as NGA50. The same behaviour was noticed between haploids of the same individual. We argue these findings give us a rough estimate of the differences between human genomes and show the importance of *de novo* assembly versus the mapping reads to the reference. Moreover, this might provide guidelines for future machine learning methods that use read mapping, including error correction, DNA modification

detection, and similar tasks. A detailed analysis of the differences between human genomes by regions can be of great importance, but this is out of the scope of this paper.

An advantage of HERRO correction is that it enables high quality using single long-read sequencing technology. It leads to lower cost, a lower amount of required genomic DNA, and simpler *de novo* assembly algorithms. By providing a model supporting R9.4.1 reads, we enable researchers to reuse old data, now with much higher accuracy.

Accurate long and ultra-long reads open many new avenues for researching more complex genomes, including those with different ploidy and aneuploidy levels, while carefully handling potentially informative positions.

Finally, the achievements detailed in this manuscript would not have been possible without the tremendous efforts of the scientific community, particularly the T2T consortium. Their production of high-quality references and provision of open access to sequenced reads were instrumental in this work.

## Methods

### Data preprocessing

Data preprocessing consists of read basecalling, adapter trimming, read splitting on middle adapters and filtering out short and low-quality reads. All datasets were basecalled using Dorado (https://github.com/nanoporetech/dorado), with the model version depending on the pore type. R9.4.1. data was basecalled using the dna_r9.4.1_e8_sup@v3.6 model, R10.4.1 4 kHz data using the dna_r10.4.1_e8.2_400bps_sup@v4.1.0 model and R10.4.1 5 kHz data using the dna_r10.4.1_e8.2_400bps_sup@v4.3.0 model. Adapter trimming is done using Porechop[42], wherein we recognize and remove adapter sets with at least one match over 95%. Next, we detect middle adapters and split reads using the *split_on_adapter* script from Duplex Tools (https://github.com/nanoporetech/duplex-tools). Dorado versions v0.5.0 or higher implement adapter trimming and splitting, making Porechop and Duplex Tools unnecessary. Finally, reads shorter than 10 kbp and those with a mean base quality below 10 are discarded using SeqKit[43,44]. Only the resulting reads will be used in the error correction process. Additional details and the preprocessing script are available in the Supplementary Information.

### Feature generation

To generate features, we first perform all-vs-all alignment using minimap2[31,45]. If multiple alignments between two reads exist, we keep only the first one, assuming that alignment is

optimal. We perform error correction on each read independently. The read that will be corrected is named *target* read, while the reads aligned with it are named *query* reads.

Initially, we divide a target read into chunks by defining non-overlapping windows of length 4096 base pairs (bp). We extract the corresponding segments from the query reads for each window based on the alignments (CIGAR string). Any query segment that doesn't cover the entire window is discarded. Moreover, a query segment for a window is also discarded if any insertion or deletion is longer than 30 bp. Next, we sort segments by alignment accuracy and retain only the top 30 segments. All segments are retained if there are fewer than 30 for a window. For each window, we define two tensors: $B \in \mathbb{R}^{31 \times L}$ and $Q \in \mathbb{R}^{31 \times L}$, corresponding to the stacked pairwise bases and stacked pairwise base qualities, respectively. The first row in both tensors corresponds to the target read, while the subsequent rows contain the sorted segments in decreasing order. If fewer than 30 segments exist, we insert a special "no alignment" token ('.') in the remaining rows.

Moreover, we define a special gap token ('*') to denote insertions with respect to the target read. Information about the alignment strand is given implicitly; bases from different strands are encoded differently. There are 11 tokens: five bases (A, C, G, T and '*') for each strand (forward and reverse) plus a no-alignment token. Base qualities for the insertions are set to 0.

In addition to features, we also extract information about potentially informative positions - supported positions. We define a position as *supported* if at least two different bases (including gap) are supported by a minimum of three reads at that specific position. Positions that do not meet this criterion are considered *unsupported*. The deep learning model predicts only the target bases from these supported positions.

**Deep learning model and training**

*Model architecture*

HERRO deep learning model consists of three parts. In the first part, we update information for each base based on its local neighbourhood before pooling information from each base at the same position to obtain a vector representation for each position. Initially, the base tensor is embedded using an embedding layer, resulting in the embedded bases $E_B \in \mathbb{R}^{31 \times L \times 6}$. This tensor and the quality base tensor are concatenated into an input tensor $I \in \mathbb{R}^{31 \times L \times 7}$, processed through two convolutional blocks. Each convolutional block consists of 1D convolution, Batch Normalization and Rectified Linear Unit (ReLU). The output of the first stage is the intermediate tensor $H^{(1)} \in \mathbb{R}^{L \times 256}$. In the second part, we focus only on representations from supported positions, updating them based on interactions with other supported positions. By concentrating only on supported positions, we process only a

fraction of all positions, drastically reducing the computational demands. The tensor $H^{(2)} \in \mathbb{R}^{S\times 256}$, containing representations of supported positions are fed into a Pre-LN Transformer encoder[46,47]. The output from the encoder is passed to the final part of the model, which consists of two classification heads: the base prediction head and the informative position prediction head. The base prediction head consists of a fully connected layer followed by a softmax layer, produces base probabilities $Y_B \in \mathbb{R}^{S\times 5}$ for every supported position. For each supported position, it predicts whether the correct base is one of the canonical bases (A, C, G, T) or a gap ('*'). The informative position prediction head, defined as a fully connected layer followed by a sigmoid, outputs the probabilities $Y_I \in \mathbb{R}^{S\times 1}$ , which denotes whether each supported position is informative. A supported position is deemed *informative* if different haplotypes have different bases at that position. The block diagram, which displays the entire architecture with defined parameters, is shown in Extended Data Fig. 2.

*Model training*

We perform multi-task learning to find the optimal model's parameters. The total loss for one window is the sum of the losses from each supported position. Total loss for supported position $i$, where $i \in \{1, 2, \dots, S\}$, is a sum of two losses: base prediction loss and informative position prediction loss and is given with :

$$L = L_B + L_I = CE\left(Y_B^i, b^i\right) + BCE\left(Y_I^i, \mathbb{1}_{b_{h1}^i \neq b_{h2}^i}\right)$$

where $CE$ is the cross-entropy loss, $BCE$ is the binary cross-entropy loss, $Y_B^i$ represents the predicted base probabilities, $b^i$ is the true base, $Y_I^i$ denotes the predicted probability that a position is informative, and $\mathbb{1}_{b_{h1}^i \neq b_{h2}^i}$ is the indicator function that equals to 1 if the bases on haplotypes $h_1$ and $h_2$ are different, and 0 otherwise.

We trained the model on 4 A100 GPUs for a maximum of 1 million steps. The batch size was 128, so the model saw 128 million examples. Validation was performed every 2000 steps, and we selected the model with the highest validation F1 score for the base prediction task. The Adam optimizer was used for training with default $\beta_1 = 0.9$ and $\beta_2 = 0.999$ parameters. We used a cosine learning rate scheduler with an initial learning rate of $3 \times 10^{-4}$. Weight decay was set to $0.1$ and gradients were clipped using the $\ell^2$-norm with a maximum norm value of $1$. The dropout value in transformer was set to $0.1$. The complete list of hyperparameters can be found in Supplementary Table 11.

*Training dataset*

We used a portion of the HG002 dataset internally sequenced by ONT for training, specifically chromosomes 1-2 and 17-18. Chromosome 20 was used for validation. The remaining chromosomes were excluded from training and model selection. The reads were first aligned to the recently published high-quality assembly using minimap2. This diploid

assembly allowed us to assign the chromosome and haplotype to each read. Only reads with primary alignments to the specified chromosomes and a mapping quality of 2 or higher were processed further, as previously described. To identify correct bases and informative positions, we mapped both contigs (haplotypes) corresponding to the assigned chromosome to each read. We extracted the correct base from the assigned haplotype for each position on a read. If the bases differed between the two haplotypes at a particular position, that position was labelled informative. If there was no mapping for the non-assigned haplotype, all positions were automatically labelled as non-informative.

**Consensus**

After obtaining predictions from the deep learning model for supported positions, we determine a consensus sequence for each window. For unsupported positions, we use simple majority voting. If two or more bases (including gaps) occur with the same frequency and one of them is a target base, we copy the target base; otherwise, the base is chosen randomly. For supported positions, we select the base with the highest predicted probability. Consecutive windows are stitched into a single sequence. If a window contains fewer than two alignments, the window is discarded, and the read is split. The resulting sequences are written to a FASTA file.

**Analysis methods**

*Generating assemblies with Hifiasm*

We ran Hifiasm v0.19.8 to generate assemblies for all corrected datasets, subsampling them to 35x coverage if they exceeded 38x. The corrected reads were chopped to 30 kbp and filtered to retain only those longer than 10 kbp. According to our experiments, this was necessary because Hifiasm did not perform optimally with longer reads as high-quality inputs. This could be related to the nature of its algorithm[34,48]. Reads that were trimmed (using porechop and duplex_tools on the fastq or trimmed during basecalling by Dorado v0.5.0 and above) but not yet corrected and that were 50 kbp or longer were used as the UL input for Hifiasm. Primary assemblies are evaluated. For diploid samples, parental data were processed using yak and used with Hifiasm's built-in trio-binning mode to obtain phased assemblies. For detailed commands, versions and parameters, see Supplementary Information, Assembly section.

*Generating assemblies with Verkko*

Verkko was run on all corrected datasets to generate assemblies, subsampling them to 35x if their coverage was 38x or above. We generally used Verkko v2.0 by default; however, in some instances where it crashed, we used a newer version (Supplementary Information, Assembly). For Verkko, the corrected reads were used as is, without chopping. Trimmed but

still needs to be corrected reads were used as the UL input for Verkko without filtering by length. For diploid samples, Meryl and Merqury were used to process parental data in Verkko's
trio-mode to generate phased assemblies. Homozygous or haploid samples were run with the "--haploid" option to improve the alignment of ONT reads. Refer to the Supplementary Information, Assembly section for detailed commands, versions and parameters.

*Generating haploid assemblies with LJA*

LJA (v0.2) was run on all homozygous or haploid corrected datasets to generate assemblies, subsampling them to 35x if they were 38x or above. Only the downsampled corrected reads were used as input for LJA. For detailed commands, see Supplementary Information, Assembly section.

*Evaluation of gene completeness*

We evaluated the gene completeness of assemblies using three different methods – asmgene, BUSCO[49,50] and compleasm[51].

For asmgene, a reference assembly was used for each sample. Minimap2 2.26-r1175 was used to align cDNA sequences to the reference assembly and then to the constructed assembly (for diploid samples, each haplotype's assembly is assessed separately). We then ran asmgene from paftools 2.26-r1175 and used its outputs to calculate how many single-copy or multi-copy genes found on the reference remained single- or multi-copy on the assembly. From this, we calculated the proportion of single-copy genes on the reference that were no longer single-copy on the assembly, the proportion of single-copy genes that became duplicated, and the proportion of multi-copy genes that were no longer multi-copy by the following:

$$Proportion\ of\ lost\ single\text{-}copy\ genes = \frac{full_sgl_{reference} - full_sgl_{assembly}}{full_sgl_{reference}}$$

$$Proportion\ of\ duplicated\ single\text{-}copy\ genes = \frac{full_dup_{assembly}}{full_sgl_{reference}}$$

$$Proportion\ of\ lost\ multi\text{-}copy\ genes = \frac{dup_cnt_{reference} - dup_cnt_{assembly}}{dup_cnt_{reference}}$$

For non-human datasets, we also additionally ran BUSCO 5.6.1 and compleasm 0.2.5 to obtain a measure of gene completeness by checking for the existence of conserved single-copy orthologs in the assembly. For detailed commands and reference for each sample, see Supplementary Information, Evaluation.

*Reference-based evaluation of assemblies*

Minigraph 0.20-r574-dirty with paftools 2.26-r1175 was used to assess each assembly's contiguity, completeness and correctness against a reference by alignment. We first aligned assembly sequences to the reference sequences using minigraph and indexed the reference with Samtools 1.19.2[52] (Using htslib[53] 1.19.1). Paftools then used this information to calculate metrics such as NG50 and NGA50. NG50 is calculated by summing the lengths of the assembled sequences in descending order, and NGA50 is similar but uses aligned segments of assembly sequences to the reference; both metrics identify the sequence length at the point where cumulative length equals or exceeds 50% of the genome length. The percentage of the reference sequences aligned to a sequence in the assemblies measures assembly completeness (Rcov; Supplementary Tables 1-8, Supplementary Table 10). Additionally, Quast 5.2.0 with minimap2 2.24-r1122 was used to evaluate T2T X and Y contigs for HG002 against the X and Y chromosomes in HG002 assembly v1.0.1. For detailed commands and the references used, see Supplementary Information, Evaluation.

*K-mers based evaluation of assemblies*

We used Merqury (with Meryl) and yak to assess assemblies using short, accurate reads from the same sample. k-mers in the short reads were counted and compared with the k-mers in assembly sequences to estimate the assembly's QV (a log-scaled measure of base error rate). For diploid samples, maternal- and paternal-specific k-mers were derived from the short reads of the parents of the assembled sample. This approach identifies k-mers from both parents in the assembled sequences, allowing it to identify switch and Hamming errors. Refer to the Supplementary information, Evaluation for detailed versions, commands and parameters.

*Counting T2T chromosomes in assemblies*

The script from https://github.com/prasad693/Tel_Sequences was used to identify telomere-to-telomere (T2T) sequences in the assemblies of all samples, except for D. melanogaster, for which telomeres do not consist of short motif repeats[54]. The script searches for species-specific canonical telomeric motifs at the ends of each sequence in the assembly and attempts to align these sequences to a reference sequence. Assembly sequences that cover the full length of the reference chromosome and also have motif repeat counts exceeding a specified threshold were considered T2T. For detailed commands, see Supplementary Information, Evaluation.

*Measurement of reads' quality*

Reads before and after correction were aligned to a reference for samples with a high-quality reference using minimap2 2.26-r1175. The bamConcordance tool was used on the

alignments to calculate $Q_{concordance}$, a score that is a log-scaled measure of base error probabilities. It also outputs the counts of different types of errors - mismatch, non-homopolymer insertion, non-homopolymer deletion, homopolymer insertion, and homopolymer deletion of the reads compared to the reference. Only the records for the primary alignments were used. Yak 0.1-r69-dirty was used to estimate QV, a log-scaled measure of base error probabilities of the reads for I002C reads corrected at different coverages. Corrected reads from HG002 (at 66x) and I002C (at 63x) were also aligned to different high-quality human assemblies to estimate the error rates of reads with respect to them. Error rates were calculated by the following:

$$Error\ rate = \frac{mismatch + insertion + deletion}{match + \ mismatch + \ insertion + deletion} * 100\%$$

$$Mismatch\ rate = \frac{mismatch}{match + mismatch + insertion + deletion} * 100\%$$

$$Indel\ rate = \frac{insertion + deletion}{match + mismatch + insertion + deletion} * 100\%$$

For detailed commands and alignment references, see Supplementary Information, Evaluation.

*Measurement of computational resources used*

The elapsed time, CPU time, and peak RAM usage were measured for the all-versus-all reads alignment, correction, and assembly steps. These numbers were taken from the output log for Hifiasm assembly runs and obtained with "/usr/bin/time—v" for the rest. GPU inference was run on the nodes of a computing cluster, where a node with 4 NVIDIA A100-SXM4-80GB was used for each run.

## Data availability

HG02818 ULONT data: https://s3-us-west-2.amazonaws.com/human-pangenomics/index.html?prefix=submissions/3471c92c-5b6a-4c20-9953-13e3a851debc--UCSC_HPRC_PLUS_nanopore_WashU_samples/HG02818/ ; HG02818 and parents' short reads data: https://s3-us-west-2.amazonaws.com/human-pangenomics/index.html?prefix=working/HPRC_PLUS/HG02818/raw_data/Illumina/ ; HG005 ONT data: https://s3-us-west-2.amazonaws.com/human-pangenomics/index.html?prefix=submissions/5b73fa0e-658a-4248-b2b8-cd16155bc157--UCSC_GIAB_R1041_nanopore/HG005_R1041_Sheared/ ; HG005 and parents' short reads data: https://s3-us-west-2.amazonaws.com/human-pangenomics/index.html?prefix=working/HPRC_PLUS/HG005/raw_data/Illumina/ ; HG002 ULONT data: https://s3-us-west-2.amazonaws.com/human-

pangenomics/index.html?prefix=submissions/5b73fa0e-658a-4248-b2b8-cd16155bc157--UCSC_GIAB_R1041_nanopore/HG002_R1041_UL/dorado/v0.4.0_wMods/ (Only the data from the four files with 'ULCIR' or 'ULNEB' in their filenames were used) ; HG002 and parents' short reads data: https://s3-us-west-2.amazonaws.com/human-pangenomics/index.html?prefix=working/HPRC_PLUS/HG002/raw_data/Illumina/ ; D. melanogaster ONT data: https://labs.epi2me.io/open-data-drosophila/ ; D. melanogaster short reads data: https://www.ncbi.nlm.nih.gov/sra/SRR6702604 ; A.thaliana ONT data: https://www.ncbi.nlm.nih.gov/sra/?term=SRR29061597 (See Supplementary Information, Additional notes on Arabidopsis thaliana data); A. thaliana Illumina data: https://ngdc.cncb.ac.cn/gsa/browse/CRA005350 ; D. rerio ULONT data: https://genomeark.s3.amazonaws.com/index.html?prefix=species/Danio_rerio/fDanRer17/genomic_data/ont/ (PAG68679 was not used for consistency because it has a different data sampling frequency from all the rest) ; D. rerio short reads data: https://www.ncbi.nlm.nih.gov/sra?linkname=bioproject_sra_all&from_uid=1029986 ; CHM13 ULONT data: https://s3-us-west-2.amazonaws.com/human-pangenomics/index.html?prefix=T2T/CHM13/nanopore/fast5/ (some partitions were skipped due to reasons such as having too many fast5 files, see Supplementary information: CHM13 data partitions used); CHM13 short reads data: https://www.ncbi.nlm.nih.gov/sra/?term=SRX1082031 ; HG002 v1.0.1: https://github.com/marbl/HG002 ; CHM13 v2.0: https://github.com/marbl/CHM13 ; I002C v.04: https://github.com/lbcb-sci/I002C or https://github.com/LHG-GG/I002C; CN1: https://github.com/T2T-CN1/CN1 ; D. rerio assembly used as reference: https://www.ncbi.nlm.nih.gov/datasets/genome/GCA_033170195.2/ ; A. thaliana assembly used as reference: https://ngdc.cncb.ac.cn/gwh/Assembly/21820/show ; D. melanogaster assembly used as reference: https://www.ncbi.nlm.nih.gov/datasets/genome/GCA_018904365.1/ ; Human cDNA used: https://ftp.ensembl.org/pub/release-111/fasta/homo_sapiens/cdna/Homo_sapiens.GRCh38.cdna.all.fa.gz ; D. melanogaster cDNA used: https://ftp.ensembl.org/pub/release-111/fasta/drosophila_melanogaster/cdna/Drosophila_melanogaster.BDGP6.46.cdna.all.fa.gz ; A. thaliana cDNA used: https://ftp.ensemblgenomes.ebi.ac.uk/pub/plants/release-58/fasta/arabidopsis_thaliana/cdna/Arabidopsis_thaliana.TAIR10.cdna.all.fa.gz ; D. rerio cDNA used: https://ftp.ensembl.org/pub/release-111/fasta/danio_rerio/cdna/Danio_rerio.GRCz11.cdna.all.fa.gz; HG002 data used for training will be available for peer-reviewers upon request; Data for I002C will be available prior to publication; HiFi reads' assembly in Table 2: https://s3-us-west-2.amazonaws.com/human-pangenomics/index.html?prefix=submissions/53FEE631-4264-4627-8FB6-09D7364F4D3B--ASM-COMP/HG002/assemblies/verkko_1.3.1/trio/ ; Duplex

reads' assembly in Table 2: https://obj.umiacs.umd.edu/marbl_publications/duplex/HG002/asms/duplex_50x_30xUL_trio.tar.gz ; Assemblies evaluated in this paper: https://zenodo.org/records/11195288 ;

## Code availability

HERRO code is available at https://github.com/lbcb-sci/HERRO. The pretrained R10.4.1 model is available at http://complex.zesoi.fer.hr/data/downloads/model_v0.1.pt and R9.4.1 model at http://complex.zesoi.fer.hr/data/downloads/model_R9_v0.1.pt.

## Acknowledgements

We would like to thank the Human Pangenome Reference Consortium (HPRC), the Telomere-to-Telomere (T2T) Consortium, the Vertebrate Genomes Project (VGP) and the Genome in a Bottle (GIAB) Consortium for making sequencing datasets publicly available. We thank George Dodd of Oxford Nanopore Technologies (ONT) for providing the Arabidopsis thaliana dataset. We thank Sean McKenzie for his assistance with datasets. We would like to acknowledge the National Genome Research Institute (NHGRI) for funding the following grants which are in support of creating the human pangenome reference: 1U41HG010972, 1U01HG010971, 1U01HG010961, 1U01HG010973, 1U01HG010963, and the Human Pangenome Reference Consortium (https://humanpangenome.org/).

This work has been jointly funded by AI Singapore and Oxford Nanopore Technologies under the grant AISG2-100E-2021-076. D. S. has also been funded by ARAP A*STAR scholarship and Croatian Science Foundation under the grant DOK-2018-09-8714.

Finally, we thank Sergey Nurk for invaluable discussions and for sharing his expertise in genome assembly.

## Author contributions

M. Š. and P. F. S. conceived the project. D. S. designed and implemented HERRO, with contributions from M. Š. and D. L. D. L. performed the assembly experiments and bioinformatics analysis, assisted by M. Š. and D. S. D. S., D. L., and M. Š. organised the manuscript. D. L., D. S., and M. Š. wrote the manuscript, with additional input from P. F. S. M. Š. and P. F. S. supervised the project, while M. Š. provided mentorship and support throughout the project.

## Competing interests

Oxford Nanopore Technologies and AI Singapore jointly funded the AI-driven De Novo Diploid Assembler project, which resulted in this manuscript. M.S. has received travel funds to speak at events hosted by Oxford Nanopore Technologies. P. F. S. is an employee of Oxford Nanopore Technologies.

## References


1. Nurk, S. *et al*. The complete sequence of a human genome. *Science* **376**, 44–53 (2022).
2. Koren, S. *et al*. Gapless assembly of complete human and plant chromosomes using only nanopore sequencing. *bioRxivorg* (2024) doi:10.1101/2024.03.15.585294.
3. Sarashetti, P., Lipovac, J., Tomas, F., Šikic, M. & Liu, J. The hitchhiker's guide to sequencing data types and volumes for population-scale pangenome construction. *bioRxiv* (2024) doi:10.1101/2024.03.14.585029.
4. Koren, S. *et al*. De novo assembly of haplotype-resolved genomes with trio binning. *Nat. Biotechnol.* (2018) doi:10.1038/nbt.4277.
5. Koren, S., Walenz, B. P., Berlin, K., Miller, J. R. & Phillippy, A. M. Canu: scalable and accurate long-read assembly via adaptive k-mer weighting and repeat separation. bioRxiv, 071282. *Genome Research* **27**, 722–736 (2017).
6. Guan, D. *et al*. Identifying and removing haplotypic duplication in primary genome assemblies. *Bioinformatics* **36**, 2896–2898 (2020).
7. Chin, C.-S. *et al*. Phased diploid genome assembly with single-molecule real-time sequencing. *Nat. Methods* **13**, 1050–1054 (2016).
8. Kolmogorov, M., Yuan, J., Lin, Y. & Pevzner, P. A. Assembly of long, error-prone reads using repeat graphs. *Nat. Biotechnol.* **37**, 540–546 (2019).

9. Shafin, K. *et al*. Haplotype-aware variant calling with PEPPER-Margin-DeepVariant enables high accuracy in nanopore long-reads. *Nat. Methods* **18**, 1322–1332 (2021).

10. Kolmogorov, M. *et al*. Scalable Nanopore sequencing of human genomes provides a comprehensive view of haplotype-resolved variation and methylation. *Nat. Methods* **20**, 1483–1492 (2023).

11. Shafin, K. *et al*. Nanopore sequencing and the Shasta toolkit enable efficient de novo assembly of eleven human genomes. *Nat. Biotechnol*. **38**, 1044–1053 (2020).

12. Nie, F. *et al*. De novo diploid genome assembly using long noisy reads. *Nat. Commun*. **15**, 2964 (2024).

13. Cheng, H., Concepcion, G. T., Feng, X., Zhang, H. & Li, H. Haplotype-resolved de novo assembly using phased assembly graphs with hifiasm. *Nat. Methods* **18**, 170–175 (2021).

14. Cheng, H. *et al*. Haplotype-resolved assembly of diploid genomes without parental data. *Nat. Biotechnol*. **40**, 1332–1335 (2022).

15. Cheng, H., Asri, M., Lucas, J., Koren, S. & Li, H. Scalable telomere-to-telomere assembly for diploid and polyploid genomes with double graph. *Nat. Methods* (2024) doi:10.1038/s41592-024-02269-8.

16. Rautiainen, M. *et al*. Telomere-to-telomere assembly of diploid chromosomes with Verkko. *Nat. Biotechnol*. **41**, 1474–1482 (2023).

17. Hu, J. *et al*. NextDenovo: an efficient error correction and accurate assembly tool for noisy long reads. *Genome Biol*. **25**, 107 (2024).

18. Darian, J. C., Kundu, R., Rajaby, R. & Sung, W.-K. Constructing telomere-to-telomere diploid genome by polishing haploid nanopore-based assembly. *Nat. Methods* **21**, 574–583 (2024).

19. Bankevich, A., Bzikadze, A. V., Kolmogorov, M., Antipov, D. & Pevzner, P. A. Multiplex de Bruijn graphs enable genome assembly from long, high-fidelity reads. *Nat. Biotechnol.* **40**, 1075–1081 (2022).

20. Rhie, A. *et al*. The complete sequence of a human Y chromosome. *Nature* **621**, 344–354 (2023).

21. Wang, T. *et al*. The Human Pangenome Project: a global resource to map genomic diversity. *Nature* **604**, 437–446 (2022).

22. Zook, J. M. *et al*. A robust benchmark for detection of germline large deletions and insertions. *Nat. Biotechnol.* **38**, 1347–1355 (2020).

23. Liao, W.-W. *et al*. A draft human pangenome reference. *Nature* **617**, 312–324 (2023).

24. Zook, J. M. *et al*. Extensive sequencing of seven human genomes to characterize benchmark reference materials. *Scientific data* vol. 3 160025 (2016).

25. Rhie, A. *et al*. Towards complete and error-free genome assemblies of all vertebrate species. *Nature* **592**, 737–746 (2021).

26. Wang, B. *et al*. High-quality Arabidopsis thaliana genome assembly with nanopore and HiFi long reads. *Genomics Proteomics Bioinformatics* **20**, 4–13 (2022).

27. Wenger, A. M. *et al*. Accurate circular consensus long-read sequencing improves variant detection and assembly of a human genome. *Nat. Biotechnol.* **37**, 1155–1162 (2019).

28. Gurevich, A., Saveliev, V., Vyahhi, N. & Tesler, G. QUAST: quality assessment tool for genome assemblies. *Bioinformatics* **29**, 1072–1075 (2013).

29. Li, H., Feng, X. & Chu, C. The design and construction of reference pangenome graphs with minigraph. *Genome Biol.* **21**, 265 (2020).

30. Rhie, A., Walenz, B. P., Koren, S. & Phillippy, A. M. Merqury: reference-free quality, completeness, and phasing assessment for genome assemblies. *Genome Biol.* **21**, 245 (2020).

31. Li, H. Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics* **34**, 3094–3100 (2018).

32. Yang, C. *et al.* The complete and fully-phased diploid genome of a male Han Chinese. *Cell Res.* **33**, 745–761 (2023).

33. Vaser, R., Sovic, I., Nagarajan, N. & Sikic, M. Fast and accurate de novo genome assembly from long uncorrected reads. *Genome Res.* gr.214270.116 (2017).

34. Kamath, S. S., Bindra, M., Pal, D. & Jain, C. Telomere-to-telomere assembly by preserving contained reads. *bioRxiv* (2023) doi:10.1101/2023.11.07.565066.

35. Šošić, M. & Šikić, M. Edlib: A C/C ++ library for fast, exact sequence alignment using edit distance. *Bioinformatics* **33**, (2017).

36. Exact global alignment using A* with chaining seed heuristic and match pruning". *Bioinformatics* (2024).

37. Kolmogorov, M. *et al.* metaFlye: scalable long-read metagenome assembly using repeat graphs. *Nat. Methods* **17**, 1103–1110 (2020).

38. Feng, X., Cheng, H., Portik, D. & Li, H. Metagenome assembly of high-fidelity long reads with hifiasm-meta. *Nat. Methods* **19**, 671–674 (2022).

39. Bertrand, D. *et al.* Hybrid metagenomic assembly enables high-resolution analysis of resistance determinants and mobile elements in human microbiomes. *Nat. Biotechnol.* **37**, 937–944 (2019).

40. Vaser, R. & Šikić, M. Time- and memory-efficient genome assembly with Raven. *Nature Computational Science* **1**, 332–336 (2021).

41. Stanojević, D., Li, Z., Foo, R. & Šikić, M. Rockfish: A Transformer-based Model for Accurate 5-Methylcytosine Prediction from Nanopore Sequencing. *bioRxiv* 2022.11.11.513492 (2022) doi:10.1101/2022.11.11.513492.

42. Wick, R. R., Judd, L. M., Gorrie, C. L. & Holt, K. E. Completing bacterial genome assemblies with multiplex MinION sequencing. *Microb. Genom.* **3**, e000132 (2017).

43. Shen, W., Sipos, B. & Zhao, L. SeqKit2: A Swiss Army Knife for Sequence and Alignment Processing. *iMeta* (2024).

44. Shen, W., Le, S., Li, Y. & Hu, F. SeqKit: A cross-platform and ultrafast toolkit for FASTA/Q file manipulation. *PLoS One* **11**, e0163962 (2016).

45. Li, H. New strategies to improve minimap2 alignment accuracy. *Bioinformatics* **37**, 4572–4574 (2021).

46. Vaswani, A. *et al.* Attention is all you need. *arXiv [cs.CL]* (2017).

47. Xiong, R. *et al.* On Layer Normalization in the Transformer Architecture. *arXiv [cs.LG]* (2020).

48. Jain, C. Coverage-preserving sparsification of overlap graphs for long-read assembly. *Bioinformatics* **39**, (2023).

49. Manni, M., Berkeley, M. R., Seppey, M., Simão, F. A. & Zdobnov, E. M. BUSCO update: Novel and streamlined workflows along with broader and deeper phylogenetic coverage for scoring of eukaryotic, prokaryotic, and viral genomes. *Mol. Biol. Evol.* **38**, 4647–4654 (2021).

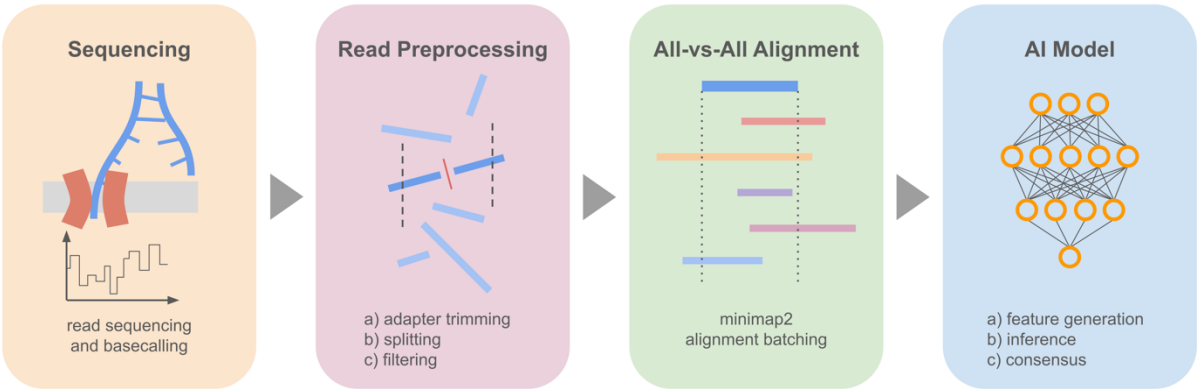
50. Manni, M., Berkeley, M. R., Seppey, M. & Zdobnov, E. M. BUSCO: Assessing genomic data quality and beyond. *Curr. Protoc.* **1**, e323 (2021).

51. Huang, N. & Li, H. compleasm: a faster and more accurate reimplementation of BUSCO. *Bioinformatics* **39**, (2023).

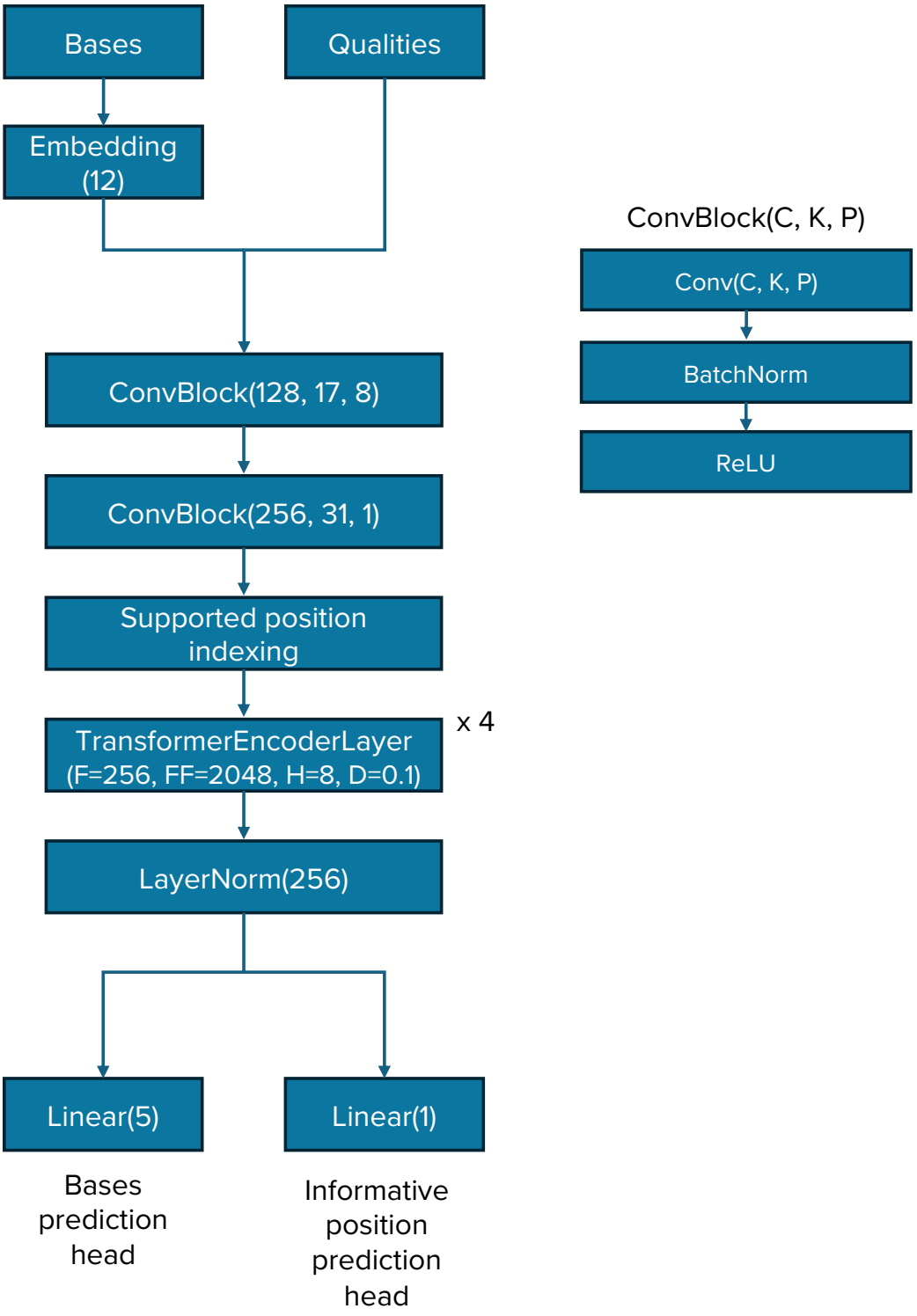
52. Danecek, P. *et al.* Twelve years of SAMtools and BCFtools. *Gigascience* **10**, (2021).

53. Bonfield, J. K. *et al.* HTSlib: C library for reading/writing high-throughput sequencing data. *Gigascience* **10**, (2021).

54. Mason, J. M. & Biessmann, H. The unusual telomeres of Drosophila. *Trends Genet.* **11**, 58–62 (1995).




*Extended Data Fig. 1* ***Overview of the HERRO method pipeline:*** *The input to the pipeline consists of reads obtained from nanopore sequencing. After trimming adapters and splitting at middle adapters, short and low-quality reads are filtered out. Next, all-vs-all alignments are performed using minimap2. In feature generation, we chunk the target read and its alignments into multiple windows. For each window, we define two matrices representing stacked bases and base qualities. These windows are then processed by a deep learning model. To predict the correct base for supported positions, we use predictions from the model. For unsupported positions, we apply simple majority voting. The final error-corrected read is obtained by stitching together the consensus sequence from each window.*




*Extended Data Fig. 2* ***HERRO Model Architecture****: The block diagram displays the complete architecture of the HERRO model, including layers and hyperparameters. The convolutional block, which consists of a convolutional layer, batch normalization, and a ReLU activation function, is fully defined on the right side of the figure.*