Verkko2 integrates proximity-ligation data with long-read De Bruijn graphs for efficient telomere-to-telomere genome assembly, phasing, and scaffolding

Dmitry Antipov,[1,4] Mikko Rautiainen,[2,4] Sergey Nurk,[3] Brian P. Walenz,[1] Steven J. Solar,[1] Adam M. Phillippy,[1] and Sergey Koren[1]

[1]Genome Informatics Section, Center for Genomics and Data Science Research, National Human Genome Research Institute, National Institutes of Health, Bethesda, Maryland 20892, USA; [2]Institute for Molecular Medicine Finland, Helsinki Institute of Life Science, University of Helsinki, Tukholmankatu 8, Biomedicum 2, Helsinki, Finland; [3]Oxford Nanopore Technologies, Oxford OX4 4DQ, United Kingdom

The Telomere-to-Telomere Consortium recently finished the first truly complete sequence of a human genome. To resolve the most complex repeats, this project relied on the semimanual combination of long, accurate Pacific Biosciences (PacBio) HiFi and ultralong Oxford Nanopore Technologies sequencing reads. The Verkko assembler later automated this process, achieving complete assemblies for approximately half of the chromosomes in a diploid human genome. However, the first version of Verkko was computationally expensive and could not resolve all regions of a typical human genome. Here we present Verkko2, which implements a more efficient read correction algorithm, improves repeat resolution and gap closing, introduces proximity-ligation-based haplotype phasing and scaffolding, and adds support for multiple long-read data types. These enhancements allow Verkko2 to assemble all regions of a diploid human genome, including the short arms of the acrocentric chromosomes and both sex chromosomes. Together, these changes increase the number of telomere-to-telomere scaffolds by twofold, reduce runtime by fourfold, and improve assembly correctness. On a panel of 19 human genomes, Verkko2 assembles an average of 39 of 46 complete chromosomes as scaffolds, with 21 of these assembled as gapless contigs. Together, these improvements enable telomere-to-telomere comparative genomics and pangenomics, at scale.

[Supplemental material is available for this article.]

Advances in sequencing technologies have set a new benchmark for de novo genome assembly: the complete assembly of entire chromosomes "from telomere to telomere" (T2T) (Nurk et al. 2022). The current sequencing recipe for achieving T2T assemblies combines "long accurate reads" (LA reads) of length >10 kb and accuracy >99.9% with "ultralong reads" (UL reads) of length >100 kb and accuracy >95%. Chromosome-scale haplotype separation is then achieved using either parental (Koren et al. 2018) or Hi-C sequencing (Cheng et al. 2021; Garg 2021) data, resulting in the complete assembly of both maternal and paternal haplotypes. Verkko (Rautiainen et al. 2023) previously demonstrated the power of this recipe, automatically assembling 20 of 46 diploid human chromosomes without gaps. Although subsequent tools have offered improved performance (Cheng et al. 2024) and a reduced dependence on trios (Lorig-Roach et al. 2024), no assembler can yet assemble all chromosomes of a typical human genome. Genomic features that remain a challenge, even for UL-read sequencing, include many functionally relevant regions of the genome such as recent segmental duplications, tandemly duplicated satellite repeats, and amplified gene arrays (e.g., rDNAs).

The short arms of the human acrocentric chromosomes, whose base-level structure was first revealed by the T2T-CHM13 reference genome (Nurk et al. 2022), are most challenging to assemble owing to their long, tandemly repeated rDNA arrays and enrichment for segmental duplications. Translocations (Ohno et al. 1961) and ongoing recombination (Guarracino et al. 2023) within the short arms have maintained a high sequence similarity across all five acrocentric chromosomes (Nurk et al. 2022). The rDNA arrays can stretch for megabases, exceeding the length of current ultralong sequencing reads. Thus, correctly resolving the acrocentrics in a diploid genome requires differentiating between 10 similar short arms and scaffolding across each rDNA array gap. Doing so is necessary for understanding variation within these important regions of the genome and better detecting abnormalities, such as Robertsonian chromosomes (de Lima et al. 2024).

The distal regions of the short arms are typically connected in the assembly graph (Fig. 1), and standard Hi-C phasing algorithms that expect a diploid "bubble chain" structure fail to separate them by haplotype (Kronenberg et al. 2021; Cheng et al. 2022). Without prior separation of chromosomes by haplotype, single-haplotype Hi-C scaffolders (Burton et al. 2013; Ghurye et al. 2019; Zhou et al. 2023) cannot be used to bridge the rDNA gaps. Furthermore, existing tools specifically designed for diploid Hi-C

[4]These authors contributed equally to this work.

Corresponding authors: adam.phillippy@nih.gov, sergey.koren@nih.gov

Article published online before print. Article, supplemental material, and publication date are at https://www.genome.org/cgi/doi/10.1101/gr.280383.124. Freely available online through the *Genome Research* Open Access option.

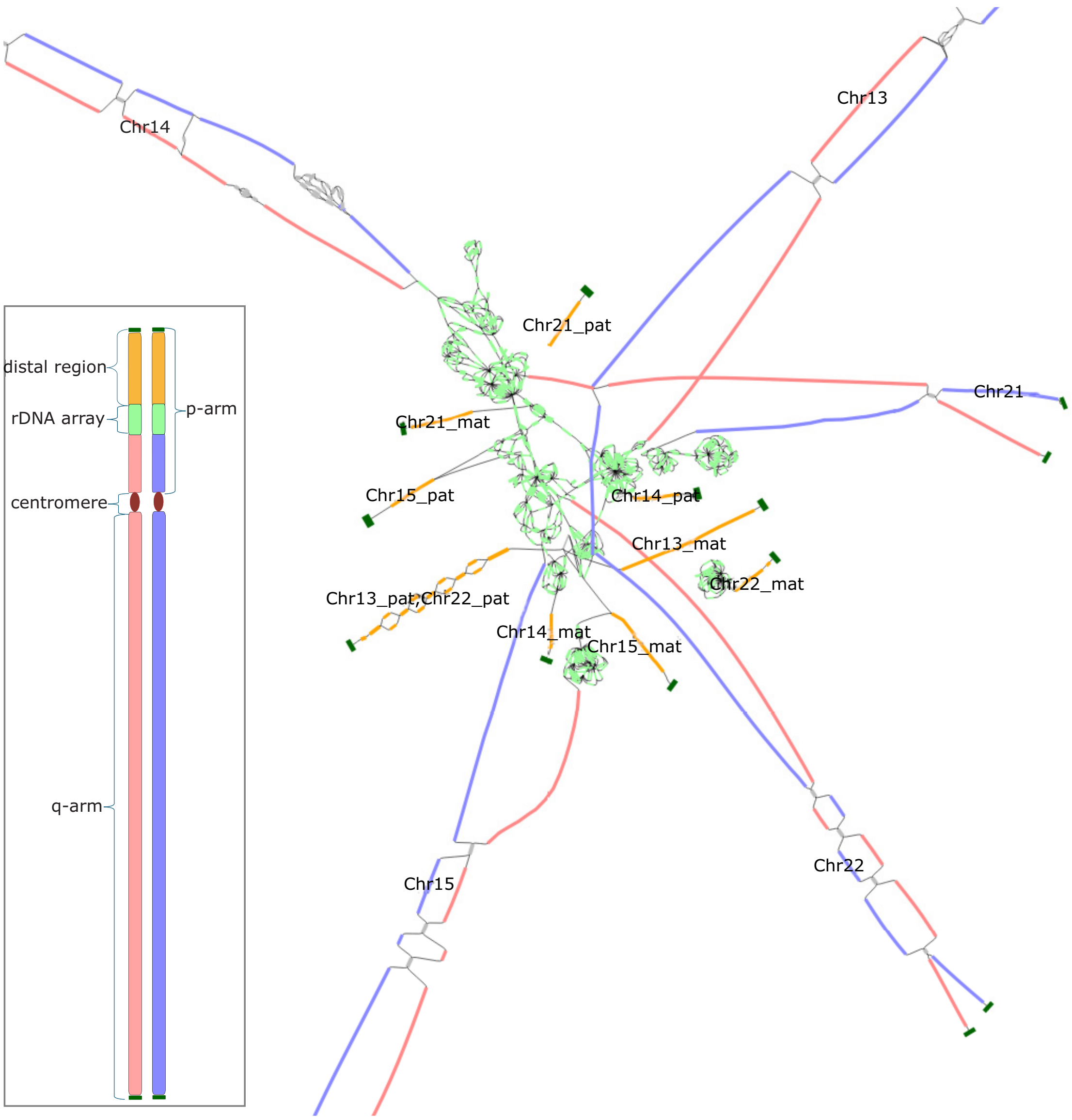

Figure 1. An assembly graph tangle resulting from the rDNA arrays of the HG002 human genome. Bandage (Wick et al. 2015) visualization of the assembly graph for the 10 HG002 acrocentric chromosomes (diploid Chromosomes 13, 14, 15, 21, 22). Maternal and paternal haplotype–assigned nodes are shown in red and blue, respectively; the rDNA repeats, in light green; the distal satellite regions, in orange; and the telomeres, in dark green. Each distal satellite is labeled according to the HG002 v1.0.1 reference assembly, with the exception of Chromosome 13 and Chromosome 22 paternal, which are too similar to be separated (Potapova et al. 2024).The *inset* shows a linear schematic of a human acrocentric chromosome with colors matching the assembly graph. The paternal and maternal haplotype–assigned sequences comprise the entire q-arm and the proximal component of the p-arm. In the contrast to other regions of the genome, the short arms of the acrocentric chromosomes do not split into a typical diploid arrangement and are difficult to phase correctly. Such complex structures violate common assumptions made by diploid phasing and scaffolding tools.

scaffolding (Zhang et al. 2019; Zeng et al. 2024) are also unsuitable for these complex and repetitive regions of the genome because they do not make use of the assembly graph or consider multi-mapped reads. This problem is not unique to human, and other species such as sheep and cattle have many more acrocentric or telocentric chromosomes than human (23 and 29, respectively) that also share a high degree of sequence similarity on their ends (Kalbfleisch et al. 2024). Enabling the T2T assembly of more samples and species requires a generalized solution for dealing with such inter-chromosomal sequence homology.

We have developed a new version of Verkko to address these issues. In addition to accelerated read-correction step, Verkko2 natively supports Hi-C data to enable the assembly of genomes for which trio information is not available. Verkko2 also introduces a new scaffolding module that is robust to the complex graph structures created by acrocentric chromosomes. Rather than scaffolding individual segments of an assembly graph (Ghurye et al. 2019), Verkko2 scaffolds haplotype paths after Hi-C or trio-based phasing. This scaffolding step considers both uniquely and multi-mapped reads, making minimal assumptions regarding the graph structure and allowing for scaffolding in even the most repetitive regions of the genome. Lastly, Verkko2 tracks every read through the assembly process and provides BAM (Li et al. 2009) output for each assembled contig, detailing the reads used, their location in the assembly, and their base-level alignment to the final sequence. This removes the need for read remapping during downstream consensus polishing and validation processes (Mc Cartney et al. 2022; Liao et al. 2023; Mastoras et al. 2025). Below, we describe these changes in detail and evaluate Verkko2 on 19 human and two non-human samples, demonstrating consistent improvements in speed, continuity, and accuracy.

Methods

The original Verkko pipeline consisted of six main steps, with module dependencies given in parentheses:

- homopolymer compression and correction of LA reads (HiCanu) (Nurk et al. 2020),
- LA multiplex De Bruijn graph construction (MBG) (Rautiainen and Marschall 2021),
- alignment of UL reads to the LA graph (GraphAligner) (Rautiainen and Marschall 2020),
- ULA multiplex De Bruijn graph construction,
- haplotype path extension using haplotype-specific markers, and
- haplotype path consensus generation from raw LA and UL reads (Canu) (Koren et al. 2017).

The ULA graph is the assembly graph incorporating the initial LA sequences followed by simplification using the UL reads (Rautiainen et al. 2023).

Our improvements to Verkko are detailed below in order of their pipeline execution. Supplemental Figure S1 includes a graphical representation of the original pipeline, adapted from Rautiainen et al. (2023), further highlighting which stages of the Verkko pipeline were modified. All evaluations and comparisons described here are based on Verkko branch v1_fxd (Verkko1) and release v2.2.1 (Verkko2). We avoid describing heuristics reused from Verkko1 and focus here on the newly added or improved features.

Long, accurate read correction

Verkko previously reused HiCanu's read correction module to correct the LA reads (Nurk et al. 2020). This step first computed all-versus-all read overlaps followed by error correction based on the resulting alignments. The initial all-versus-all computation was a performance bottleneck and used as much as 79% of total runtime for the HG002 human genome assembly (Rautiainen et al. 2023). Verkko2 introduces a more efficient overlapper to reduce runtime. The previous method was originally designed for Sanger (Sanger et al. 1977) data and thus was limited to small k-mers, did not use minimizers, and extended single k-mer short seeds (Myers et al. 2000). Together this made it increasingly expensive as read length and error increased. The current method is similar to previous minimizer-based approaches (Roberts et al. 2004; Berlin et al. 2015; Li 2016) with the addition of HPC and simple-sequence compression, as done in MBG (Rautiainen and Marschall 2021).

The algorithm used by the new overlapper is based on long-k minimizer matching (Roberts et al. 2004; Jain et al. 2018). First, the reads are homopolymer- and microsatellite-compressed (Rautiainen et al. 2023) to mask systematic sequencing errors. Then, minimizers are extracted from each read using a 64-bit rolling hash and minimizer parameters of k-mer length $k = 425$ and window size $w = 19$. The sequences of the minimizers are ignored, and only the hash value and start and end positions within the read are stored. A maximum copy count (default, 1000) is used to filter out repetitive minimizers. Matches per read are found based on the hash values and positions. The same minimizer occurring in two different reads implies a match between the reads, with the start positions of the minimizers specifying the match diagonal. The minimizer matches are sorted by diagonal and clustered such that any diagonal difference of at least 500 bp between adjacent matches separates them into different clusters. Each cluster is then considered one overlap, with the start and end positions of the overlap taken from the minimum and maximum start and end positions of the minimizers in the cluster. A minimum overlap length threshold (default, 1000 bp) is used to remove spurious overlaps. Although the use of a 64-bit hash can introduce spurious matches, it is unlikely to cause spurious overlaps because that would require two or more hash collisions on similar match diagonals.

Given n reads with l base pairs each, the minimizer parameters k and w, the number of temporary hash files f, and the number of distinct hashes with copy count at least two h, the index construction takes the worst case $O(nl)$ time and $O\left(\frac{nl}{w}\right)$ temporary disk space, disk writes, and reads and produces an index file of size $O\left(\frac{nl}{w}\right)$. The expected index construction memory use is $O\left(\frac{nl}{wf} + h\right)$, where the first term is the memory use for counting hash multiplicity and assigning IDs to the hashes, and the second term is the memory used for keeping the mapping from hashes to IDs. The value of h is typically small and close to genome size divided by w because it is composed of k-mers that appear at least twice in the reads, excluding most spurious k-mers.

To find the overlaps between the reads, the index file is iterated in multiple batches. Once all-versus-all overlaps are computed, Verkko reuses the previous code to compute base-level alignments and extend each overlap beyond the k-mer matches.

Details of the overlapper implementation are available in Supplemental Methods S1.

Repeat identification

As shown by Rautiainen et al. (2023), Verkko performs well for the majority of the genome. However, some regions were prone to issues in Verkko1. The first improvement was to tolerate coverage fluctuations in the input LA sequencing reads. Both HiFi and ONT Duplex sequences, which are the most commonly used LA data types, have biases that lead to reduced coverage in certain contexts (Nurk et al. 2020; Koren et al. 2024). A critical step for repeat resolution is the initial identification of repeat and unique nodes by coverage to create source and sink nodes for subsequent ONT-based resolution. This list is further refined using graph structure and UL-read alignments (Rautiainen et al. 2023), but no nodes can be added. Therefore, the coverage-based criteria need to capture all possibly unique nodes. Verkko1 identified an initial set of unique nodes by comparing the region's coverage to the global average, allowing more deviation from average for larger nodes (50 kb) than short ones because they are less likely to be repetitive. Verkko2 keeps the node-length dependence but instead compares the coverage against the local connected component. It also allows a larger decrease (60% instead of 30%) in coverage for the large nodes. Together, these changes better resolve regions with systematically lower or higher coverage than Verkko1.

Telomere assembly

Another region Verkko1 struggled with was the telomere or the terminal node in a connected component in the graph. In many cases, assemblies would extend up to the telomere but miss the terminal node. Two issues were primarily responsible for this loss of sequence. First, tip clipping could remove multiple tip nodes during simplification, leaving no telomere sequence in the graph. Instead, we now ensure at least one tip node is retained, preferring to keep the highest coverage or, when all nodes are equal coverage, the longest node. Second, the identification of unique nodes did not flag the last node in a component as unique because these nodes frequently have lower than expected coverage or are short. This prevented subsequent simplification from reaching the telomere, leaving an ambiguous branch in the graph. Verkko2 relaxes the uniqueness criteria for dead-end nodes to ensure that these branches are resolved.

Gap closing

While investigating gaps in Verkko1 assemblies, we found that GraphAligner could systematically misalign reads to the wrong haplotype in cases in which one haplotype had missing sequence, owing to a lack of LA coverage. The correct alignment requires splitting a read into at least two pieces and leaving bases unaligned. As a result, common scoring matrices within aligners like GraphAligner and minimap2 (Li 2021) prefer the more complete, but wrong, haplotype (Supplemental Fig. S2), leaving unresolved gaps in the assembly. To address this, we use a twofold strategy of (1) improvements to GraphAligner and (2) supplemental alignments from Winnowmap (Jain et al. 2022).

First, GraphAligner was modified to use a new diploid heuristic for read alignment. This heuristic is based on finding homologous pairs of nodes and the heterozygous *k*-mers that distinguish them, matching the read to a specific haplotype based on matching heterozygous *k*-mers and preventing the read from aligning to nodes in the other haplotype. Despite the name, the diploid heuristic does not specifically use haplotype information and instead deals with alignment in repeats with copy count two in general, which most commonly are the two haplotypes of a diploid genome. This section uses the terms *heterozygous* to refer to any unique, one-copy sequence in the genome and *homozygous* for any two-copy sequence.

To detect heterozygous *k*-mers, first all closed syncmers (Edgar 2021) are found and counted from the node sequences with parameters *k* and *s* and are used as the *k*-mers for the rest of the diploid heuristic. Any heterozygous *k*-mer must have a copy count of exactly one. However, a *k*-mer with copy count of one may not necessarily be a heterozygous *k*-mer. This includes cases in which one of the haplotypes has a gap owing to low LA coverage or assembly issues, in which case any homozygous *k*-mers in the remaining haplotype will have a copy count of one in the graph.

Homozygous, two-copy *k*-mers are used to identify homology between different nodes. After the *k*-mer copy counts are tabulated, GraphAligner detects substrings that are enclosed by two-copy *k*-mers on both ends and contain only one-copy *k*-mers in between. If two such substrings are found in different nodes such that the enclosing two-copy *k*-mers match, the substrings are considered to be a homologous pair. Because the substrings are enclosed by *k*-mers with copy count of two, any such substring can match with at most one other substring, but because the nodes are of variable length, multiple short nodes can be homologous to a single long node. All homologous pairs are detected in the graph and stored in an index. In addition, the heterozygous *k*-mers inside the homologous substrings, as well as their locations, are stored as a list of haplotype informative *k*-mers.

When aligning a read, the diploid heuristic is used to decide which nodes should be used based on their *k*-mer matches. Haplotype informative *k*-mers are found in the read and clustered based on their match diagonal between the read and the nodes. If at least three haplotype informative *k*-mers from the same node are found within a 100 bp diagonal, the node is considered to match the read at the corresponding location. All other nodes that are homologous to the matched node are also checked. If there is even one haplotype informative *k*-mer match to an another homologous node, then the pair is considered ambiguous and ignored. If there are no haplotype informative *k*-mer matches to the other node (i.e., it appears to be from a different "haplotype"), the read is forbidden from aligning to the unmatched node in the region covered by the matching node.

In the seed-and-extend phase of alignment, when a read is forbidden from aligning to a node, any seed hits found between the read and the forbidden node are discarded, and the dynamic programming algorithm is not allowed to extend the alignment into a forbidden node. Importantly, the alignment is only forbidden within the region covered by the matching node, because inexact two-copy repeats within the same haplotype often have homologous substrings, and forbidding the alignment over the entire read would cause issues in such regions.

The diploid algorithm requires syncmer parameters *k* and *s*. Higher values of *k* will find more haplotype informative *k*-mers in repetitive areas and require higher accuracy reads in order to find matches, whereas a lower *k* will find fewer haplotype informative *k*-mers but enable the heuristic to work with higher error reads. GraphAligner allows using multiple *k* values that will run the diploid heuristic separately for each *k* and then merge the lists of forbidden nodes. Parameter *s* is always set to five, and *k* defaults to $k=21$ and $k=31$ for the two rounds of the diploid heuristic. Future higher accuracy UL reads may benefit from higher values of *k*.

Lastly, Winnowmap alignments are used as a supplement to the GraphAligner alignments, because Winnowmap was designed for haplotype-specific alignments (but is not graph-aware). In this phase, all tips in the graph are identified. Such nodes can arise from either sequencing errors or coverage gaps. For each tip, we identify if it has an alternate complete haplotype by considering nontip paths originating from the tip's predecessor node. Any reads aligned by GraphAligner to these nodes are then extracted and realigned to the full graph using Winnowmap. The merged GraphAligner and Winnowmap alignments are input to the same gap filling procedure used in Verkko1. Reads not used for gap filling in this step have their alignments reverted to the original GraphAligner alignments to allow for their use in other graph simplification operations.

Hi-C data processing

In contrast with prior Hi-C phasing approaches, Verkko2 does not rely on a max-cut optimization (Cheng et al. 2022; Lorig-Roach et al. 2024) to separate bubble structures in the graph. Instead, homozygous nodes are identified via coverage, and a modified Kernighan–Lin (Kernighan and Lin 1970) algorithm is used to partition the nodes into two components.

Verkko2 uses Hi-C chromosome conformation capture data for both phasing and scaffolding of the assembly graph and supports both short-read Hi-C as well as Nanopore-based Pore-C (Deshpande et al. 2022) sequencing data. Critically, Verkko integrates this long-range linkage information directly with the assembly graph, improving accuracy over standalone phasing and scaffolding tools.

The two main features that make Hi-C data useful for T2T assembly are that (1) Hi-C read pairs are more frequently derived from the same haplotype than between different haplotypes, and (2) Hi-C read pairs are more frequently derived from close rather than distant positions in the genome. These features are based on the 3D organization of the genome and make Hi-C data particularly useful for phasing (Selvaraj et al. 2013) and scaffolding (Burton et al. 2013), respectively.

Figure 2 summarizes the new steps added to Verkko pipeline to enable Hi-C-based phasing and scaffolding of Verkko's ULA assembly graph. Hi-C read pairs are first aligned to the assembled unitigs with BWA-MEM (Li 2013). All read pairs in which at least one read is mapped ambiguously (mapping quality of zero, typically multimappers from homozygous regions of different haplotypes) are ignored for phasing but used later in scaffolding. For the alignment of the long-read Pore-C data, we use minimap2 (Li 2021) and consider each pair of fragments in a read separately. Otherwise, the handling of Pore-C data mirrors that of Hi-C.

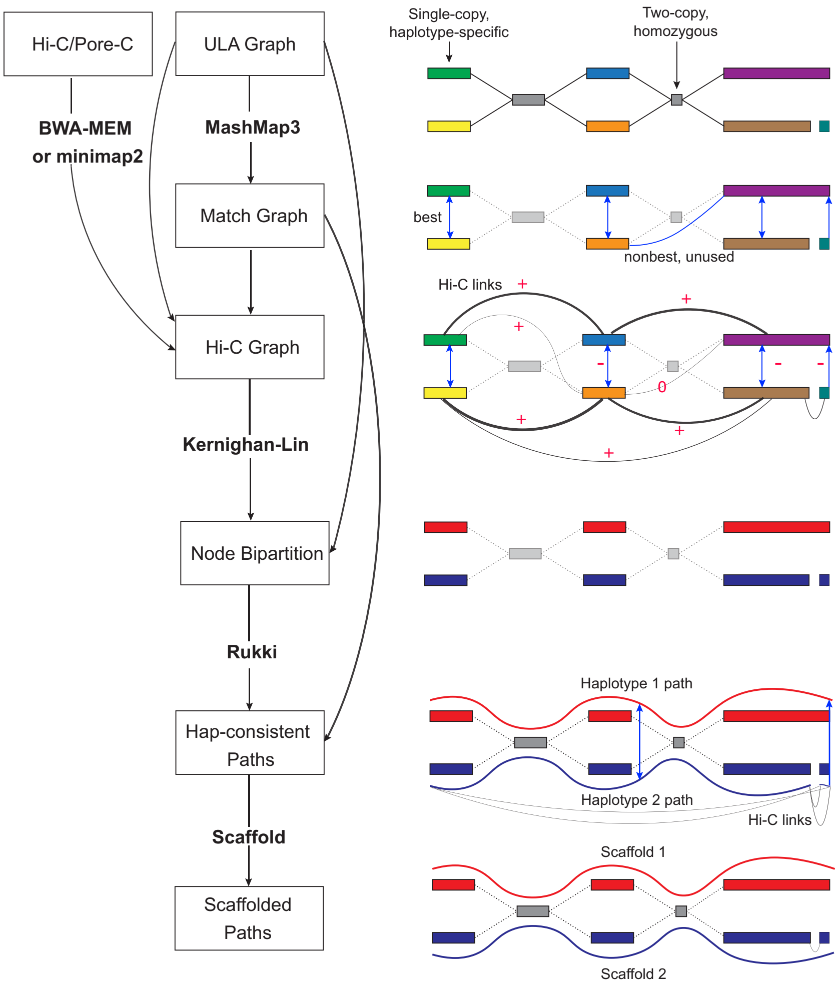

Figure 2. Overview of the Verkko2 Hi-C/Pore-C processing. The process starts with the ULA graph built from the LA and UL sequences. Note that the ULA graph links are only used to cluster the graph into connected components so they are shown as dotted gray lines in the figure. The BWA-MEM/minimap2 step aligns the Hi-C or Pore-C data to the sequences of the ULA graph nodes, and counts connecting pairs of nodes are tallied. Next, the Match Graph step ignores homozygous (based on coverage) and short (by default ≤200 kb) nodes. The remaining nodes are self-aligned to identify homology. The initially computed Hi-C edges are filtered using the Match Graph to build the Hi-C Graph. In many cases, the highest-count Hi-C connection is between homologous pairs of nodes. To avoid these false links, edges connecting potentially repetitive nodes are removed (shown with a value of zero), whereas edges connecting homologous nodes are given large negative weights (shown with –). These updated link weights are used to bipartition each connected component of the graph into two haplotypes. These partitions are then provided to Rukki along with the ULA graph to generate haplotype paths, and the pipeline proceeds as in Verkko1. These haplotype-consistent paths are again used to identify homology, shown with blue edges based on the Match Graph step. All four possible connections are considered to connect the two blue paths based on Hi-C link evidence. In this example, the alignment of both blue paths to the red path adds a multiplicative bonus to one Hi-C connection consistent with the alignment, leading to a scaffold connecting the blue paths.

Hi-C haplotype phasing

Because Hi-C read pairs more often originate from the same haplotype, the frequency of Hi-C links between assembly graph nodes can be used to phase (i.e., partition) the graph by haplotype. However, because of Hi-C library noise, sequencing errors, and mapping issues, homologous nodes from different haplotypes can also share a high number of Hi-C links. If only considering the frequency of Hi-C links between nodes, this can pose a problem for phasing.

To resolve the issue of Hi-C links between homologous nodes, we construct an additional set of edges on top of the ULA graph that are based on sequence similarity between the nodes, called the Match Graph. Homologous nodes in the ULA graph are identified by an all-versus-all similarity search using MashMap3 (Kille et al. 2023), and all pairs of nodes with at least one homology match over a minimum length threshold (default, 200 kb) are connected by an edge in the Match Graph. The weight of each edge is set to the total bases covered by alignments between the corresponding nodes.

One might expect that a node would be connected only to the corresponding homologous node of similar size. However, this is not the case in real assembly graphs, because some regions may be assembled in one node in one haplotype but split into multiple nodes in the other. Worse, some relatively long regions between heterologous chromosomes can share sequences that are similar in identity to that of homologous chromosomes, for example, centromeres, rDNAs, pseudoautosomal regions (PARs), and pseudohomologous regions (PHRs) (Nurk et al. 2022; Guarracino et al. 2023). To address this issue, the edges of the Match Graph are processed in two different ways to distinguish homology between *haplotypes* and *repeats*. For each node *V* in the Match Graph, we check whether there is an edge incident to *V* with the weight being a clear majority (default > 1.5 ∗ second_best_weight) among all edges incident to *V* and also greater than one-third of the length of at least one of those nodes. Such edges are considered to link a pair of haplotypes, whereas edges of the Match Graph that are not best for either of their incident nodes are considered to link a pair of repeats.

Hi-C read pair alignments are then considered along with the Match Graph to build the Hi-C Graph. The Hi-C Graph is constructed from long nodes (default > 200 kb) of the ULA graph, with undirected, weighted edges corresponding to the number of Hi-C read pairs mapping between a pair of nodes. To correct for erroneous Hi-C links between homologous nodes, nodes connected by a haplotype edge in the Match Graph are given a large negative weight (default $-10*$maximum_edge_weight) in the Hi-C Graph, and to correct for mismapped Hi-C pairs between repeats, nodes connected by a repeat edge in the Match Graph are removed from the Hi-C Graph. Additionally, self-edges are removed because they carry no additional phasing information.

The resulting Hi-C Graph is then phased by identifying a bipartition of the nodes that minimizes the total weight of edges between two similarly sized subsets. Although this problem is known to be NP-complete (Garey 1997), the Kernighan–Lin algorithm (Kernighan and Lin 1970) is a commonly used greedy heuristic implemented by the NetworkX Python library (Hagberg et al. 2008). It operates by iteratively swapping pairs of nodes that yield the largest decrease in edge weight between the two partitions. We begin with multiple (default, 1000) randomly generated bipartitions and perform 10,000 iterations of the Kernighan–Lin algorithm for each component. Afterward, any short nodes not included in the initial Hi-C Graph are assigned based on whether a clear majority ($best \geq 2.5 * second\_best$) of Hi-C links connects it to one of the two subgraphs.

Although it is possible to phase the full Hi-C Graph at once, we split it into subgraphs that are expected to represent diploid chromosome components prior running the Kernighan–Lin algorithm and process each individually. These components are defined by supplementing the ULA assembly graph with the haplotype edges of the Match Graph, after which any Hi-C link-connected nodes from different components are ignored. This helps to reduce the number of iterations of Kernighan–Lin and enables the detection and proper handling of heterogametic sex chromosomes. After phasing, the bipartitions of each component are arbitrarily merged to generate a single bipartition of the entire ULA graph, which is then passed to the Rukki module to identify haplotype paths in the same manner as a trio-based bipartition (Rautiainen et al. 2023). Supplemental Figure S3 summarizes all steps of the Hi-C phasing process in detail.

Haplotype path reconstruction

Rukki is an independent Verkko module for extracting haplotype paths from a labeled assembly graph. Given an assembly graph and node labels corresponding to haplotype information (either trio or Hi-C based), Rukki classifies the nodes into maternal, paternal, or homozygous categories and then performs a heuristic search for haplotype paths starting from long nodes in the graph. The homozygous nodes are assumed to be sequences of perfect similarity between the two haplotypes and are included in both. The sequences of the resulting haplotype paths can be extracted from the graph, forming contigs (gapless) and scaffolds (gapped).

In Verkko2, Rukki has been improved to minimize the number of nodes that remain unassigned to a haplotype. First, large nodes (default, >500 kb) with expected coverage for a single haplotype are assigned by simple majority rather than requiring a minimum ratio (default, 5:1) of one haplotype to the other. Second, all nodes in a tangle, in which the tangle is connected to only a single haplotype, are assigned to the same haplotype. Third, in simple bubbles in which the nodes are too short to be assigned by the first criterion and/or in which insufficient signal exists for haplotype separation, Rukki now forces the two paths of the bubble into different haplotypes. Previously, Rukki used a single path through the bubble for both haplotypes, which resulted in a loss of heterozygosity. Rukki relies on any available phasing signal to assign the nodes, with random assignment if no haplotype information is available. Lastly, Rukki now provides a text description for the causes of scaffold gaps, including graph tangles, coverage dropouts, or ambiguous bubbles with inadequate phasing information.

Hi-C scaffolding

In addition to using Hi-C for phasing, Verkko2 introduces the use of Hi-C data for scaffolding to connect haplotype paths that remain broken owing to missing edges in the assembly graph (e.g., owing to low sequencing coverage) or separated in the graph by very large tangles (e.g., owing to the rDNA arrays). Scaffolding is performed directly on the haplotype-specific paths identified by Rukki in the preceding step.

Hi-C interaction strengths between each pair of haplotype paths are computed based on Hi-C read pairs mapped within a fixed-length prefix/suffix of the paths (default, 5 Mb or half the path length, whichever is smaller), and scores are computed for all for possible orientations of the pair (fwd-fwd, fwd-rev, rev-rev, rev-fwd). To account for multimapping reads, we filter out read pairs that have at least one read with more than five mappings of equivalent quality or both reads in the pair that have at least one mapping to the same graph node.

The scaffolding step considers two types of Hi-C link weights: a *primary* weight using all read pairs including the remaining multimappers normalized by their number of alignments (i.e., if one read in a pair is mapped to two places and the second read is mapped to three, each link is counted with weight 1/6) and a *secondary* weight using only uniquely mapped reads. Additional scaffolding information contained within the assembly graph is incorporated via multiplicative bonuses on the link weights. First, paths that are close (closer than 500 kb) in the assembly graph are more likely to be consecutive, and their corresponding link weights are multiplied by a constant factor (default, three). Second, it is common for a sequencing gap to be present in only one of the two homologous chromosomes, so using the previously constructed Match Graph, we check if paths p_1 and p_2 are homologous haplotypes of some third path p_3 and if the position of these matches are close in p_3. If so, a constant multiplicative bonus (default, five) is applied. This homology bonus can also be applied using alignments to a reference genome, as a form of reference-assisted scaffolding.

Given the Hi-C link weights as computed above, scaffolds are constructed via the connection of mutually best pairs of paths. For simplicity, both orientations of a haplotype path are considered independently but forbidden from appearing in the same scaffold. For each path p_1 that does not end with a telomere, we check for the existence of a clear best forward extension path p_2 and also require that p_1 is a clear backward extension for p_2 (best$>$ $1.5*$second_best). If there is no mutual extension using the primary weights, we check the secondary weights using the same rule.

Lastly, subtelomeric repeat complexity and sequencing dropout can lead to scaffolds that are missing a small telomere-containing node at their ends. To resolve such cases, we perform telomere-informed graph scaffolding in cases in which one haplotype is T2T but the homologous haplotype appears broken. Given an assembly graph component in which one Hi-C path p_1 contains telomere on both ends and the homologous path p_2 contains telomere only on one end, if (1) there is a path p_t that also contains telomere on one end, (2) p_t is near (closer than 500 kb) p_2 in the graph, and (3) no other paths containing nodes from this component >1 Mb, then p_2 and p_t are scaffolded together.

Assembly postprocessing

The output of the scaffolding module is an updated set of paths through the ULA graph, which are converted to sequences through the existing Verkko consensus procedure (Rautiainen et al. 2023). Verkko2 adds the option to generate a BAM file describing the positions and alignments of the underlying reads.

These alignments are informed by knowledge contained within the assembly pipeline, such as when LA read *k*-mers were used to construct nodes and when UL reads were aligned to the graph, resulting in a more accurate representation than if the reads were naively mapped to the resulting assembly. This output is useful for both inspecting the quality of the resulting consensus sequence and for potentially further improving the quality of the consensus through raw read, signal-level polishing (Mastoras et al. 2025).

To facilitate easier submission to the public sequence archives, Verkko2 also includes support for the removal of small, unassembled sequences, as well as the isolation of specific, user-defined sequences. For human assemblies, default screening sequences are provided for mitochondrial, rDNA, and Epstein–Barr virus (EBV) sequences (NC_012920 [Andrews et al. 1999], KY962518 [Kim et al. 2018], and AJ507799 [Arrand et al. 1981], respectively), but there are no restrictions on what sequences can be used, and this feature can also be used for the removal of any known contaminant.

To screen the final assembly, first, nodes with no connection to any other node in the graph are identified and removed if shorter than a minimum length (default, <100 kb). Second, all sequences in the assembly are compared against the screening sequences using MashMap3. Mapping results are filtered to remove those with an estimated match identity <97.5% or those that cover <10% of the sequence. If the remaining hits, combined, cover at least half of the sequence(s), it is declared a match and removed from the assembly. Additionally, any node within four hops of a match is also declared a match if its sequence is not more than fourfold longer than the matched node sequence, by default. This captures more diverged versions of the screened sequence and excludes incomplete resolutions of repetitive arrays unless they are included in a large node. For all matched contigs, an exemplar sequence is reported as the sequence with the largest breadth of coverage by matches or as the match with the highest LA read coverage when multiple contigs exist with matches covering >90% of their length. Finally, because circular molecules (e.g., mitochondria) are often assembled into linear contigs with redundant sequence on their ends (Hunt et al. 2015), the extracted exemplar is trimmed to remove any redundancy >1 kb.

Results

Benchmarking

We selected two human data sets from the Human Pangenome Reference Consortium (HPRC) (Liao et al. 2023) for detailed benchmarking. We selected data sets with both Hi-C and parental trio data available for comparison to allow trio-based validation. Although newer data are available for HG002, we used the same data that was used by Rautiainen et al. (2023) to highlight the algorithmic improvements rather than sequencing technology evolution. To measure Verkko's performance on species other than human, we included a sheep cross (of *Ovis canadensis* and *Ovis aries*) from the Ruminant T2T Consortium (Kalbfleisch et al. 2024; Olagunju et al. 2024) and a chicken from the Vertebrate Genomes Project (Rhie et al. 2021). Supplemental Table S1 summarizes the read and genome information for all included data sets.

We report the number of T2T contigs and scaffolds for measuring continuity; the hamming and switch error rates for phasing correctness estimated with yak (https://github.com/lh3/yak); the Phred-scaled quality value (QV) for consensus accuracy measured with yak; and compleasm (Huang and Li 2023) for completeness at the gene level with the haplotypes evaluated independently, corrected for sex chromosome genes, and values averaged between the haplotypes. Hamming and switch error rates are estimates of long- and short-range phasing errors. Hamming error is the proportion of incorrectly phased, haplotype-specific *k*-mers in a scaffold. Switch error is the number of changes from one haplotype-specific *k*-mer to the other within a scaffold. Missing and duplicated genes are counted against the OrthoDB odb10 databases (Manni et al. 2021) for primates, Cetartiodactyla, and Aves, respectively, which contain genes expected to be present exactly once in each haplotype in all representatives of the order. T2T contigs and scaffolds are both defined by the presence of telomeric sequence on the ends but with scaffolds allowing for gaps (typically caused by large tandem repeat arrays). Comparison of Verkko2's trio-based and Hi-C-based phasing are shown in Table 1, demonstrating comparable accuracy between the two methods. To avoid overcounting the number of missing genes for heterogametic samples, missing genes were also counted for the autosomal sequences as identified by MashMap3 alignment to the existing reference genomes (CHM13 for human [Nurk et al. 2022], GCA_024206055.1 [Huang et al. 2023] for chicken, and GCF_016772045.2 [Davenport et al. 2022] for sheep).

We next compared Verkko2 against Verkko1 and Hifiasm (Cheng et al. 2021), which was recently updated to support hybrid assembly with UL reads (version 0.19.8) (Cheng et al. 2024). Figure 3 summarizes key metrics with the full results provided in Supplemental Table S2. Verkko2 shows clear improvements in T2T scaffolds over both tools, with similar correctness results, and a reduced runtime compared with Verkko1. Verkko2 is now also faster than Hifiasm, but with a runtime dependent on the sequencing coverage and repeat structure of each genome. Hamming error rate results are similar with the exception of Hifiasm Hi-C, which is less accurate than other tools. Focusing on the distal regions of the human acrocentric chromosomes, Verkko2 Hi-C resolves five and six out of the 10 possible acrocentric chromosomes for HG002 and HG00733, respectively, whereas no other assemblies resolve more than one. In the sheep *O. aries* haplotype, in which the majority of chromosomes are telocentric and share sheep satellite I repeats at one end (D'Aiuto et al. 1997), Verkko1 resolved only four chromosomes, including Chromosomes 3 and X, which are known to be deficient in this satellite (Burkin et al. 1996). In contrast, Verkko2 Hi-C resolved an additional six T2T scaffolds for this challenging *O. aries* haplotype. In the chicken genome, Verkko2 had fewer T2T scaffolds compared with Hifiasm. However, chicken, and birds in general, suffer from a drop-out of HiFi coverage on the microchromosomes (Huang et al. 2023). Because of its string graph approach, Hifiasm is more tolerant of low-coverage HiFi data, leading to improved continuity. Hifiasm is also more aggressive in assigning contigs to haplotypes, improving the missing gene score on the chicken sample at the expense of phasing error (Supplemental Table S2). To show structural correctness, we have generated Hi-C contact maps for both haplotypes of all Verkko2 Hi-C assemblies used in Table 1 with curationpretext (Supplemental Figs. S4–S7; Ewels et al. 2020; https://github.com/sanger-tol/curationpretext).

The above reference-free metrics demonstrate Verkko2's improvements for both human and non-human genomes, but none of these metrics are able to capture large, structural errors in the assemblies. Existing reference-free tools such as Flagger (Liao et al. 2023) and NucFrec (Vollger et al. 2019) produce a high number of false-positive predictions and are not always consistent with each other (Porubsky et al. 2025), making it challenging to use them without manual curation of their results. A high-

Table 1. Verkko2 Hi-C versus trio benchmarking

	Sheep	Chicken	HG002	HG00733
T2T scf				
Hi-C	**31**	**34**	**40**	**41**
Trio	23	32	32	33
T2T ctg				
Hi-C	**24**	21	21	**26**
Trio	20	**25**	**22**	23
Hamming error				
Hi-C	**0.85%**	**0.58%**	**0.39%**	**0.75%**
Trio	0.88%	0.60%	**0.39%**	0.77%
Switch error				
Hi-C	**0.95%**	**0.13%**	**0.41%**	**0.79%**
Trio	**0.95%**	**0.13%**	**0.41%**	**0.79%**
QV				
Hi-C	**54.17**	45.13	53.87	**53.86**
Trio	**54.17**	**45.17**	**53.89**	53.82
Missing genes				
Hi-C	1.37%	3.12%	1.61%	**0.09%**
Trio	**1.36%**	**2.52%**	**1.60%**	**0.09%**
Missing genes (no sex chrs)				
Hi-C	**0.06%**	1.06%	**0.09%**	**0.09%**
Trio	**0.06%**	**0.44%**	**0.09%**	**0.09%**
Dup genes				
Hi-C	1.73%	**0.12%**	**0.68%**	**0.71%**
Trio	**1.72%**	0.22%	0.69%	0.72%

T2T contigs/scaffolds were counted as those >5 Mb with telomeres on both ends. Scaffolds were broken into contigs at any stretch of more than three N's, and contigs/scaffolds <100 kb were discarded for all metrics. Verkko Hi-C shows comparable QV, switch, hamming, and missing gene stats to Verkko trio, but it consistently has a higher count of T2T scaffolds owing to its ability to restore missing connectivity using Hi-C links. Bold text indicates the best score for each metric and species; ties are also bolded.

quality T2T reference genome exists for HG002, but although reference-based validation (Gurevich et al. 2013) is commonly used to evaluate the quality of haploid genomes, there is not yet a standard tool for evaluating the quality of large, diploid assemblies. We use QUAST (Gurevich et al. 2013; Mikheenko et al. 2018), which although not specifically designed for diploid assemblies, is suitable to perform a relative comparison of base-level and structural correctness. For all metrics reported, Verkko2 trio is the most accurate, with Verkko2 Hi-C a close second (Supplemental Table S3). Verkko2 also reduces misassemblies by almost twofold versus Verkko1 and has about 2.5-fold fewer base level errors compared with Hifiasm.

HPRC year 1 reassembly

To evaluate Verkko2's performance across a larger set of samples, we selected a subset of 17 HPRC samples (Liao et al. 2023) for which both trio and Hi-C data were available (Fig. 4; Supplemental Table S4). These Verkko2 HPRC assemblies have a median T2T scaffold count of 39, median hamming and switch error rates <0.6%, and a median QV of 54.94. The core gene statistics can identify both missing or misassigned sequences. For example, if homologous sequences are placed into the same haplotype, it would have an increase in duplicated genes, whereas the other haplotype would have an increase in missing genes. If only one haplotype has an increase in missing genes, the sequence for it is likely incomplete but was not placed in the other haplotype. However, the assemblies are stable with a median missing gene rate of 0.17%. One sample (HG02559) was assembled into 45 T2T scaffolds, one short of the ideal 46 for a diploid human genome, whereas 40% of all samples have more than 40 T2T scaffolds. We confirmed the absence of chimeric assemblies by aligning all Verkko scaffolds ≥500 kb to the T2T-CHM13 reference genome (Nurk et al. 2022) using MashMap3 and found no scaffold with >1% of its sequence mapped to more than one chromosome (excluding the ChrX/Y PAR and acrocentric chromosomes; see below). In comparison, the published HPRCv1 assemblies from Liao et al. (2023), which did not incorporate the UL data, achieved only two T2T scaffolds (compared with 666 for Verkko2), median hamming and switch error rates of 0.7% and 0.6%, median missing genes per haplotype of 0.23%, and a median QV of 53.6 (Liao et al. 2023). Thus, Verkko2's incorporation of UL and Hi-C data not only improves T2T assembly continuity but also reduces overall assembly errors.

This large set of human trio samples also provided the opportunity to further measure our Hi-C scaffolding accuracy, specifically within the difficult-to-validate short arms of the acrocentric chromosomes. These short arms share regions of sequence similarity >99% identity between heterologous chromosomes (Nurk et al. 2022) and undergo meiotic recombination (Guarracino et al. 2023), confounding sequence alignment and ruling out reference-based validation. Additionally, these chromosomes are known to colocalize within nucleoli, potentially violating our assumptions of Hi-C interaction patterns. To validate Verkko2's scaffolding performance within these difficult regions, we relied on the trio information to identify haplotype switches in the acrocentric scaffolds. Although this evaluation will not detect all scaffolding errors (e.g., *maternal* Chromosome 13 short arm swapped with *maternal* Chromosome 14 short arm), we would expect a random scaffold join within the acrocentrics to introduce a haplotype switch with ∼50% probability (e.g., *maternal* Chromosome 13 short arm swapped with *paternal* Chromosome 13 short arm). Using Merqury (Rhie et al. 2020), we detected the parental origin (maternal/paternal/unassigned) for each long node (>1 Mb) and quantified haplotype switches between the distal region of the short arm and the rest of the acrocentric chromosome (spanning the rDNA array). In total, Verkko2 scaffolded 93 acrocentric chromosomes and introduced no haplotype switches between the distal region and the rest of the chromosome. A single switch, occurring within the distal region, was observed on a short arm of sample HG03453. Thus, the absence of haplotype switches observed across the rDNA array within these 17 HPRC samples suggests that our Hi-C scaffolding method is generally accurate. For the HG002 genome, in which distal regions have been experimentally validated with fluorescence in situ hybridization (Potapova et al. 2024), Verkko2 scaffolded five distal regions, which were all consistent with the reference.

Discussion

The Verkko2 improvements described here improve the number and quality of T2T chromosomes assembled, while simultaneously reducing computational requirements. Combined, these changes approach the goal of complete, diploid T2T assembly (e.g., 46

T2T chromosomes per human genome) (see Supplemental Table S4). For the most difficult to assemble human acrocentric chromosomes, Verkko2 was able to resolve 55% of all chromosomes tested as T2T scaffolds with only the rDNA array typically remaining as a gap. Assembling across these most repetitive regions of the genome will enable future research on their unique recombination patterns (Guarracino et al. 2023) and will improve our understanding of the evolutionary history and structural variation within these important chromosomes across large pangenome data sets such as the HPRC (Liao et al. 2023).

To simplify sequencing requirements, Verkko2 now supports multiple long-read data types, in addition to HiFi, for initial construction of the LA multiplex De Bruijn graph. Verkko2 results on ONT Duplex data and precorrected ONT Simplex are previously described (Koren et al. 2024; Sarashetti et al. 2024; Stanojevic et al. 2024). Verkko2 now also provides base-level alignments and location information for all reads used to generate the assembly. This can be combined with existing validation and polishing tools (Mc Cartney et al. 2022; Liao et al. 2023; Mastoras et al. 2025) to incorporate the assembler's representation of the genome and is a first step toward routinely reporting quality values for all bases in the assembly.

Areas for future improvement include better utilization of depth of coverage information, improved assemblies in the absence of UL data, pseudohaplotype outputs in the absence of phasing data, and support for polyploid genomes. Although Verkko2 is more tolerant to coverage fluctuations, coverage is still not used in the most optimal way. Verkko2 only uses coverage from uniquely placed LA nodes, leading to systematic underestimates in repetitive regions. Improved coverage estimates, specifically the identification of unique nodes and the estimated copy number of repeat nodes, would improve overall assembly continuity. Second, Verkko's pipeline is currently optimized for the hybrid combination of LA and UL reads. In cases in which only LA data are available, graph simplification steps currently only performed on the ULA graph could be performed directly on the LA graph, improving results in the absence of UL data. Third, Verkko2 only outputs completely phased contigs and thus produces more fragmented assemblies when no long-range information (such as trio, Hi-C, or Strand-seq) (Henglin et al. 2024) is available. In this case, a pseudohaplotype graph traversal that allows for phase switches, while preserving the overall structure of the assembly, would improve continuity. Verkko2's scaffolding could be further improved by considering multiple suffix/prefix lengths rather than a single fixed value, as is done in YAHS (Zhou et al. 2023), considering multiway interactions from Pore-C libraries rather than only pairwise and extending the phasing/scaffolding methods to polyploid genomes. Fourth, Verkko2 relies on heuristics to simplify the assembly graph and incorporate Hi-C data. Although designed to be sequence agnostic, applying Verkko2 to the new types of sequencing technologies and/or genomes may break these assumptions and require updated parameters. In the future, the increased availability of high-quality T2T reference genomes could enable more optimized assembly parameterization with machine learning.

The routine assembly of complete genomes promises to improve all aspects of genomics. For example, the current paradigm for human clinical genomics involves mapping reads against a single reference genome and identifying the variants between the sample and reference. However, this approach is fundamentally flawed owing to reference bias and limits our understanding of the genome, especially in the context of multicopy gene families and other repetitive regions. Inferring the complete, diploid genome of an individual, at a cost that is comparable to current mapping-based approaches, promises to correct this deficiency (Ebler et al. 2022). Until then, Verkko2's input flexibility, fast runtime,

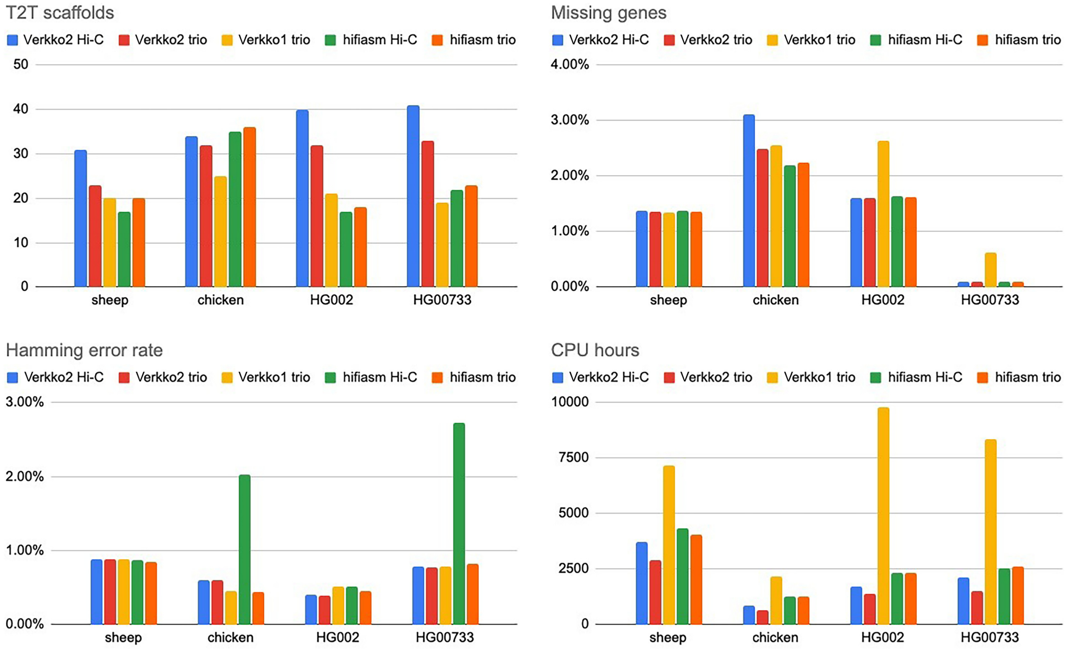

Figure 3. Comparison of tested assemblers with all statistics measured as before. Verkko Hi-C has the highest T2T scaffold rate (except on chicken), followed by Verkko2 trio. Both versions of Hifiasm are comparable to Verkko1. With the exception of Hifiasm Hi-C on chicken and HG00733, all assemblers have comparable hamming error rates. Verkko1 has a consistently higher rate of missing genes compared with other assemblies. All assemblers were run on the NIH Biowulf compute cluster.

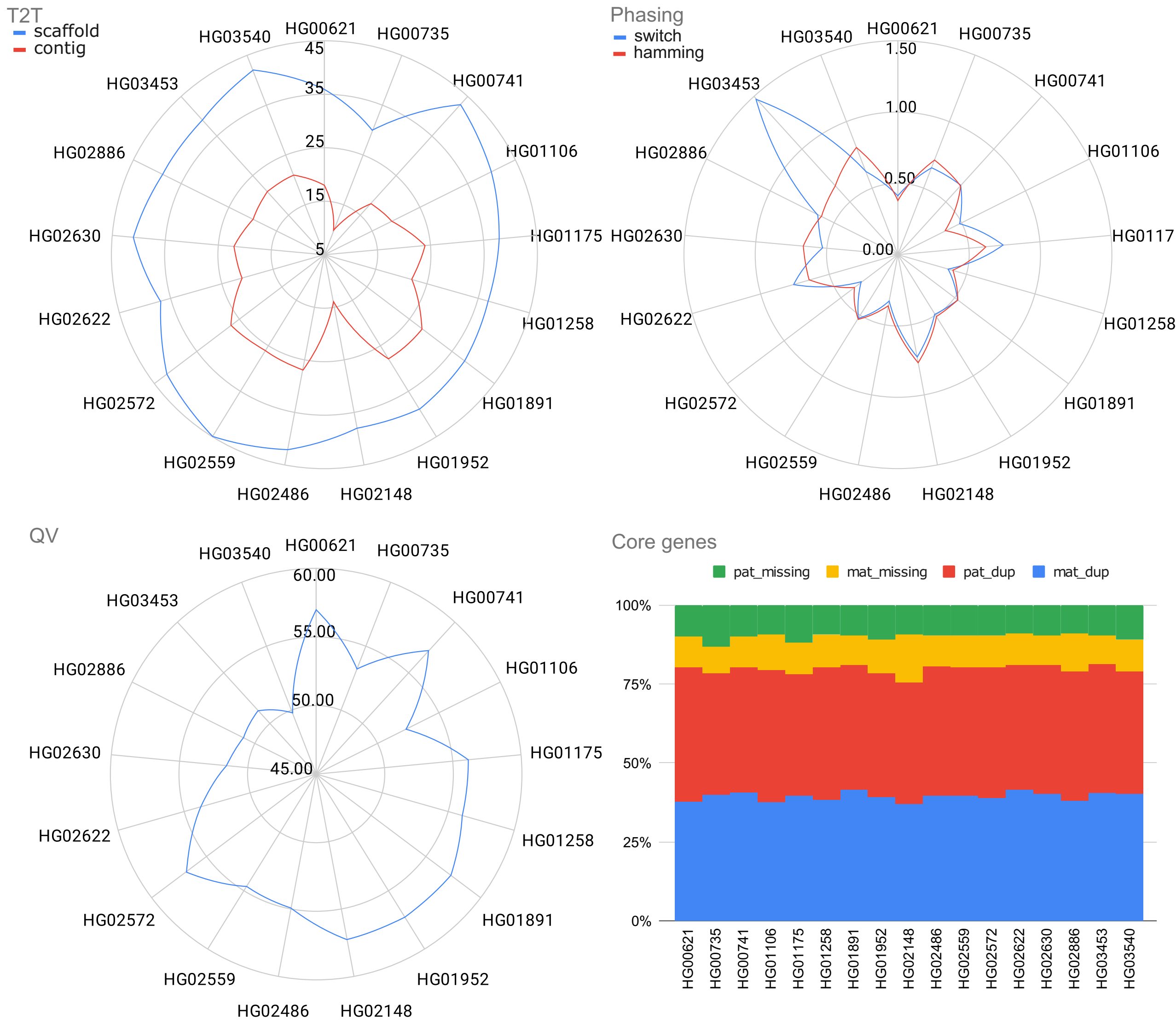

Figure 4. Verkko2 results on the HPRC year 1 assemblies. The T2T statistics are computed as before. QV and phasing error are computed using yak, and the average of both haplotypes is reported. The core genes are computed using compleasm and reported for each haplotype. The dup categories report single-copy genes present more than once in a haplotype. Because of natural variation, a small number (<1%) of duplicated genes is expected. The missing gene categories report single-copy genes not present in the assembly, excluding the sex chromosomes. The stability of duplicated and missing genes across all samples supports that Verkko2 is accurately reconstructing the full sequence for both haplotypes.

and novel scaffolding methods enable the routine assembly of personalized reference genomes and the construction of a comprehensive T2T pangenome database. This will expand the fraction of the genome and types of complex variation that are captured by large-scale genomic studies and surveyed by clinical diagnostic tests, bringing us one step closer to a future of personalized T2T genomics.

Data set availability

No new data were generated for this study. Sheep reads were generated by the ruminant T2T consortium (Kalbfleisch et al. 2024). Reads for chicken sample were downloaded from https://www.genomeark.org/genomeark-all/Gallus_gallus.html for individual bGalGal1 and its parents bGalGal2 and bGalGal3. All reads for human samples were downloaded from HPRC's Amazon cloud at https://s3-us-west-2.amazonaws.com/human-pangenomics/index.html.

Software availability

Verkko2 is available on GitHub (https://github.com/marbl/verkko). Version tagged v2.2.1 used for benchmarking in this study is available at GitHub (https://github.com/marbl/verkko/releases/tag/v2.2.1) and as Supplemental Code.

Competing interest statement

S.K. has received travel funds to speak at events hosted by Oxford Nanopore Technologies. S.N. is an employee of Oxford Nanopore Technologies. The remaining authors declare no competing interests.

Acknowledgments

We thank the HPRC assembly working group and Shilpa Garg for helpful discussions on Hi-C alignment and phasing methods. We thank the Ruminant T2T Consortium for sharing the bighorn sequencing data and Wesley Warren (U.S. Department of A National Institute of Food and Agriculture [NIFA] 2022-67015-36218), Erich Jarvis, and the Vertebrate Genome Project for sharing the chicken data. This work was supported, in part, by the Intramural Research Program of the National Human Genome Research Institute, National Institutes of Health (NIH) (D.A., B.P.W., S.J.S., A.M.P., and S.K.). This work utilized the computational resources of the NIH HPC Biowulf cluster (https://hpc.nih.gov) and CSC-IT Center for Science Finland.

Author contributions: D.A., M.R., S.N., B.P.W., S.J.S., and S.K. developed and implemented algorithmic improvements. D.A. developed the new scaffolding module and performed benchmarking. M.R. developed the new overlapper module. S.K. supervised software development. A.M.P. directed the project. D.A., S.K., and A.M.P. drafted the manuscript with methods contributions from M.R., S.N., and S.J.S. All authors read and approved the final manuscript.

References

Andrews RM, Kubacka I, Chinnery PF, Lightowlers RN, Turnbull DM, Howell N. 1999. Reanalysis and revision of the Cambridge reference sequence for human mitochondrial DNA. *Nat Genet* **23:** 147. doi:10.1038/13779

Arrand JR, Rymo L, Walsh JE, Björck E, Lindahl T, Griffin BE. 1981. Molecular cloning of the complete Epstein-Barr virus genome as a set of overlapping restriction endonuclease fragments. *Nucleic Acids Res* **9:** 2999–3014. doi:10.1093/nar/9.13.2999

Berlin K, Koren S, Chin C-S, Drake JP, Landolin JM, Phillippy AM. 2015. Assembling large genomes with single-molecule sequencing and locality-sensitive hashing. *Nat Biotechnol* **33:** 623–630. doi:10.1038/nbt.3238

Burkin DJ, Broad TE, Jones C. 1996. The chromosomal distribution and organization of sheep satellite I and II centromeric DNA using characterized sheep-hamster somatic cell hybrids. *Chromosome Res* **4:** 49–55. doi:10.1007/BF02254945

Burton JN, Adey A, Patwardhan RP, Qiu R, Kitzman JO, Shendure J. 2013. Chromosome-scale scaffolding of de novo genome assemblies based on chromatin interactions. *Nat Biotechnol* **31:** 1119–1125. doi:10.1038/nbt.2727

Cheng H, Concepcion GT, Feng X, Zhang H, Li H. 2021. Haplotype-resolved de novo assembly using phased assembly graphs with hifiasm. *Nat Methods* **18:** 170–175. doi:10.1038/s41592-020-01056-5

Cheng H, Jarvis ED, Fedrigo O, Koepfli K-P, Urban L, Gemmell NJ, Li H. 2022. Haplotype-resolved assembly of diploid genomes without parental data. *Nat Biotechnol* **40:** 1332–1335. doi:10.1038/s41587-022-01261-x

Cheng H, Asri M, Lucas J, Koren S, Li H. 2024. Scalable telomere-to-telomere assembly for diploid and polyploid genomes with double graph. *Nat Methods* **21:** 967–970. doi:10.1038/s41592-024-02269-8

D'Aiuto L, Barsanti P, Mauro S, Cserpan I, Lanave C, Ciccarese S. 1997. Physical relationship between satellite I and II DNA in centromeric regions of sheep chromosomes. *Chromosome Res* **5:** 375–381. doi:10.1023/A:1018444325085

Davenport KM, Bickhart DM, Worley K, Murali SC, Salavati M, Clark EL, Cockett NE, Heaton MP, Smith TPL, Murdoch BM, et al. 2022. An improved ovine reference genome assembly to facilitate in-depth functional annotation of the sheep genome. *GigaScience* **11:** giab096. doi:10.1093/gigascience/giab096

de Lima LG, Guarracino A, Koren S, Potapova T, McKinney S, Rhie A, Solar SJ, Seidel C, Fagen B, Walenz BP, et al. 2024. The formation and propagation of human Robertsonian chromosomes. bioRxiv doi:10.1101/2024.09.24.614821

Deshpande AS, Ulahannan N, Pendleton M, Dai X, Ly L, Behr JM, Schwenk S, Liao W, Augello MA, Tyer C, et al. 2022. Identifying synergistic high-order 3D chromatin conformations from genome-scale nanopore concatemer sequencing. *Nat Biotechnol* **40:** 1488–1499. doi:10.1038/s41587-022-01289-z

Ebler J, Ebert P, Clarke WE, Rausch T, Audano PA, Houwaart T, Mao Y, Korbel JO, Eichler EE, Zody MC, et al. 2022. Pangenome-based genome inference allows efficient and accurate genotyping across a wide spectrum of variant classes. *Nat Genet* **54:** 518–525. doi:10.1038/s41588-022-01043-w

Edgar R. 2021. Syncmers are more sensitive than minimizers for selecting conserved *k*-mers in biological sequences. *PeerJ* **9:** e10805. doi:10.7717/peerj.10805

Ewels PA, Peltzer A, Fillinger S, Patel H, Alneberg J, Wilm A, Garcia MU, Di Tommaso P, Nahnsen S. 2020. The nf-core framework for community-curated bioinformatics pipelines. *Nat Biotechnol* **38:** 276–278. doi:10.1038/s41587-020-0439-x

Garey MR. 1997. *Computers and intractability: a guide to the theory of NP-completeness*. W.H. Freeman, New York.

Garg S. 2021. Computational methods for chromosome-scale haplotype reconstruction. *Genome Biol* **22:** 101. doi:10.1186/s13059-021-02328-9

Ghurye J, Rhie A, Walenz BP, Schmitt A, Selvaraj S, Pop M, Phillippy AM, Koren S. 2019. Integrating Hi-C links with assembly graphs for chromosome-scale assembly. *PLoS Comput Biol* **15:** e1007273. doi:10.1371/journal.pcbi.1007273

Guarracino A, Buonaiuto S, de Lima LG, Potapova T, Rhie A, Koren S, Rubinstein B, Fischer C, Abel HJ, et al. 2023. Recombination between heterologous human acrocentric chromosomes. *Nature* **617:** 335–343. doi:10.1038/s41586-023-05976-y

Gurevich A, Saveliev V, Vyahhi N, Tesler G. 2013. Quast: quality assessment tool for genome assemblies. *Bioinformatics* **29:** 1072–1075. doi:10.1093/bioinformatics/btt086

Hagberg A, Swart P, Chult DS. 2008. *Exploring network structure, dynamics, and function using network*, technical report. Los Alamos National Laboratory (LANL), Los Alamos, NM.

Henglin M, Ghareghani M, Harvey WT, Porubsky D, Koren S, Eichler EE, Ebert P, Marschall T. 2024. Graphasing: phasing diploid genome assembly graphs with single-cell strand sequencing. *Genome Biol* **25:** 265. doi:10.1186/s13059-024-03409-1

Huang N, Li H. 2023. compleasm: a faster and more accurate reimplementation of BUSCO. *Bioinformatics* **39:** btad595. doi:10.1093/bioinformatics/btad595

Huang Z, Xu Z, Bai H, Huang Y, Kang N, Ding X, Liu J, Luo H, Yang C, Chen W, et al. 2023. Evolutionary analysis of a complete chicken genome. *Proc Natl Acad Sci* **120:** e2216641120. doi:10.1073/pnas.2216641120

Hunt M, Silva ND, Otto TD, Parkhill J, Keane JA, Harris SR. 2015. Circlator: automated circularization of genome assemblies using long sequencing reads. *Genome Biol* **16:** 294. doi:10.1186/s13059-015-0849-0

Jain C, Dilthey A, Koren S, Aluru S, Phillippy AM. 2018. A fast approximate algorithm for mapping long reads to large reference databases. *J Comput Biol* **25:** 766–779. doi:10.1089/cmb.2018.0036

Jain C, Rhie A, Hansen NF, Koren S, Phillippy AM. 2022. Long-read mapping to repetitive reference sequences using Winnowmap2. *Nat Methods* **19:** 705–710. doi:10.1038/s41592-022-01457-8

Kalbfleisch TS, McKay SD, Murdoch BM, Adelson DL, Almansa-Villa D, Becker G, Beckett LM, Benítez-Galeano MJ, Biase F, Casey T, et al. 2024. The Ruminant Telomere-to-Telomere (RT2T) Consortium. *Nat Genet* **6:** 1566–1573. doi:10.1038/s41588-024-01835-2

Kernighan BW, Lin S. 1970. An efficient heuristic procedure for partitioning graphs. *Bell Syst Tech J* **49:** 291–307. doi:10.1002/j.1538-7305.1970.tb01770.x

Kille B, Garrison E, Treangen TJ, Phillippy AM. 2023. Minmers are a generalization of minimizers that enable unbiased local Jaccard estimation. *Bioinformatics* **39:** btad512. doi:10.1093/bioinformatics/btad512

Kim J-H, Dilthey AT, Nagaraja R, Lee H-S, Koren S, Dudekula D, Wood WH III, Piao Y, Ogurtsov AY, Utani K, et al. 2018. Variation in human chromosome 21 ribosomal RNA genes characterized by TAR cloning and long-read sequencing. *Nucleic Acids Res* **46:** 6712–6725. doi:10.1093/nar/gky442

Koren S, Walenz BP, Berlin K, Miller JR, Bergman NH, Phillippy AM. 2017. Canu: scalable and accurate long-read assembly via adaptive *k*-mer weighting and repeat separation. *Genome Res* **27:** 722–736. doi:10.1101/gr.215087.116

Koren S, Rhie A, Walenz BP, Dilthey AT, Bickhart DM, Kingan SB, Hiendleder S, Williams JL, Smith TP, Phillippy AM. 2018. De novo assembly of haplotype-resolved genomes with trio binning. *Nat Biotechnol* **36:** 1174–1182. doi:10.1038/nbt.4277

Koren S, Bao Z, Guarracino A, Ou S, Goodwin S, Jenike KM, Lucas J, McNulty B, Park J, Rautiainen M, et al. 2024. Gapless assembly of complete human and plant chromosomes using only nanopore sequencing. *Genome Res* **34:** 1919–1930. doi:10.1101/gr.279334.124

Kronenberg ZN, Rhie A, Koren S, Concepcion GT, Peluso P, Munson KM, Porubsky D, Kuhn K, Mueller KA, Low WY, et al. 2021. Extended haplotype-phasing of long-read de novo genome assemblies using Hi-C. *Nat Commun* **12:** 1935. doi:10.1038/s41467-020-20536-y

Li H. 2013. Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. arXiv:1303.3997 [q-bio.GN]. doi:10.48550/arXiv.1303.3997

Li H. 2016. Minimap and miniasm: fast mapping and de novo assembly for noisy long sequences. *Bioinformatics* **32:** 2103–2110. doi:10.1093/bioinformatics/btw152

Li H. 2021. New strategies to improve minimap2 alignment accuracy. *Bioinformatics* **37:** 4572–4574. doi:10.1093/bioinformatics/btab705

Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, Marth G, Abecasis G, Durbin R, 1000 Genome Project Data Processing Subgroup. 2009. The Sequence Alignment/Map format and SAMtools. *Bioinformatics* **25:** 2078–2079. doi:10.1093/bioinformatics/btp352

Liao W-W, Asri M, Ebler J, Doerr D, Haukness M, Hickey G, Lu S, Lucas JK, Monlong J, Abel HJ, et al. 2023. A draft human pangenome reference. *Nature* **617:** 312–324. doi:10.1038/s41586-023-05896-x

Lorig-Roach R, Meredith M, Monlong J, Jain M, Olsen HE, McNulty B, Porubsky D, Montague TG, Lucas JK, Condon C, et al. 2024. Phased nanopore assembly with Shasta and modular graph phasing with GFAse. *Genome Res* **34:** 454–468. doi:10.1101/gr.278268.123

Manni M, Berkeley MR, Seppey M, Simão FA, Zdobnov EM. 2021. 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. doi:10.1093/molbev/msab199

Mastoras M, Asri M, Brambrink L, Hebbar P, Kolesnikov A, Cook DE, Nattestad M, Lucas J, Won TS, Chang P-C, et al. 2025. Highly accurate assembly polishing with DeepPolisher. *Genome Res* (this issue) **35:** 1595–1608. doi:10.1101/gr.280149.124

Mc Cartney AM, Shafin K, Alonge M, Bzikadze AV, Formenti G, Fungtammasan A, Howe K, Jain C, Koren S, Logsdon GA, et al. 2022. Chasing perfection: validation and polishing strategies for telomere-to-telomere genome assemblies. *Nat Methods* **19:** 687–695. doi:10.1038/s41592-022-01440-3

Mikheenko A, Prjibelski A, Saveliev V, Antipov D, Gurevich A. 2018. Versatile genome assembly evaluation with QUAST-LG. *Bioinformatics* **34:** i142–i150. doi:10.1093/bioinformatics/bty266

Myers EW, Sutton GG, Delcher AL, Dew IM, Fasulo DP, Flanigan MJ, Kravitz SA, Mobarry CM, Reinert KH, Remington KA, et al. 2000. A whole-genome assembly of *Drosophila*. *Science* **287:** 2196–2204. doi:10.1126/science.287.5461.2196

Nurk S, Walenz BP, Rhie A, Vollger MR, Logsdon GA, Grothe R, Miga KH, Eichler EE, Phillippy AM, Koren S. 2020. HiCanu: accurate assembly of segmental duplications, satellites, and allelic variants from high-fidelity long reads. *Genome Res* **30:** 1291–1305. doi:10.1101/gr.263566.120

Nurk S, Koren S, Rhie A, Rautiainen M, Bzikadze AV, Mikheenko A, Vollger MR, Altemose N, Uralsky L, Gershman A, et al. 2022. The complete sequence of a human genome. *Science* **376:** 44–53. doi:10.1126/science.abj6987

Ohno S, Trujillo J, Kaplan W, Kinosita R, Stenius C. 1961. Nucleolus-organisers in the causation of chromosomal anomalies in man. *Lancet* **278:** 123–126. doi:10.1016/S0140-6736(61)92647-2

Olagunju TA, Rosen BD, Neibergs HL, Becker GM, Davenport KM, Elsik CG, Hadfield TS, Koren S, Kuhn KL, Rhie A, et al. 2024. Telomere-to-telomere assemblies of cattle and sheep Y-chromosomes uncover divergent structure and gene content. *Nat Commun* **15:** 8277. doi:10.1038/s41467-024-52384-5

Porubsky D, Dashnow H, Sasani TA, Logsdon GA, Hallast P, Noyes MD, Kronenberg ZN, Mokveld T, Koundinya N, Nolan C, et al. 2025. Human de novo mutation rates from a four-generation pedigree reference. *Nature* doi:10.1038/s41586-025-08922-2

Potapova T, Kostos P, McKinney S, Borchers M, Haug J, Guarracino A, Solar S, Gogol M, Anez GM, de Lima LG, et al. 2024. Epigenetic control and inheritance of rDNA arrays. bioRxiv doi:10.1101/2024.09.13.612795

Rautiainen M, Marschall T. 2020. GraphAligner: rapid and versatile sequence-to-graph alignment. *Genome Biol* **21:** 253. doi:10.1186/s13059-020-02157-2

Rautiainen M, Marschall T. 2021. MBG: minimizer-based sparse de Bruijn Graph construction. *Bioinformatics* **37:** 2476–2478. doi:10.1093/bioinformatics/btab004

Rautiainen M, Nurk S, Walenz BP, Logsdon GA, Porubsky D, Rhie A, Eichler EE, Phillippy AM, Koren S. 2023. Telomere-to-telomere assembly of diploid chromosomes with Verkko. *Nat Biotechnol* **41:** 1474–1482. doi:10.1038/s41587-023-01662-6

Rhie A, Walenz BP, Koren S, Phillippy AM. 2020. Merqury: reference-free quality, completeness, and phasing assessment for genome assemblies. *Genome Biol* **21:** 245. doi:10.1186/s13059-020-02134-9

Rhie A, McCarthy SA, Fedrigo O, Damas J, Formenti G, Koren S, Uliano-Silva M, Chow W, Fungtammasan A, Kim J, et al. 2021. Towards complete and error-free genome assemblies of all vertebrate species. *Nature* **592:** 737–746. doi:10.1038/s41586-021-03451-0

Roberts M, Hayes W, Hunt BR, Mount SM, Yorke JA. 2004. Reducing storage requirements for biological sequence comparison. *Bioinformatics* **20:** 3363–3369. doi:10.1093/bioinformatics/bth408

Sanger F, Nicklen S, Coulson AR. 1977. DNA sequencing with chain-terminating inhibitors. *Proc Natl Acad Sci* **74:** 5463–5467. doi:10.1073/pnas.74.12.5463

Sarashetti P, Lipovac J, Tomas F, Šikic M, Liu J. 2024. The hitchhiker's guide to sequencing data types and volumes for population-scale pangenome construction. bioRxiv doi:10.1101/2024.03.14.585029

Selvaraj S, Dixon JR, Bansal V, Ren B. 2013. Whole-genome haplotype reconstruction using proximity-ligation and shotgun sequencing. *Nat Biotechnol* **31:** 1111–1118. doi:10.1038/nbt.2728

Stanojević D, Lin D, Florez De Sessions P, Šikić M. 2024. Telomere-to-telomere phased genome assembly using error-corrected simplex nanopore reads. bioRxiv doi:10.1101/2024.05.18.594796

Vollger MR, Dishuck PC, Sorensen M, Welch AE, Dang V, Dougherty ML, Graves-Lindsay TA, Wilson RK, Chaisson MJ, Eichler EE. 2019. Long-read sequence and assembly of segmental duplications. *Nat Methods* **16:** 88–94. doi:10.1038/s41592-018-0236-3

Wick RR, Schultz MB, Zobel J, Holt KE. 2015. Bandage: interactive visualization of de novo genome assemblies. *Bioinformatics* **31:** 3350–3352. doi:10.1093/bioinformatics/btv383

Zeng X, Yi Z, Zhang X, Du Y, Li Y, Zhou Z, Chen S, Zhao H, Yang S, Wang Y, et al. 2024. Chromosome-level scaffolding of haplotype-resolved assemblies using Hi-C data without reference genomes. *Nat Plants* **10:** 1184–1200. doi:10.1038/s41477-024-01755-3

Zhang X, Zhang S, Zhao Q, Ming R, Tang H. 2019. Assembly of allele-aware, chromosomal-scale autopolyploid genomes based on Hi-C data. *Nat Plants* **5:** 833–845. doi:10.1038/s41477-019-0487-8

Zhou C, McCarthy SA, Durbin R. 2023. YaHS: yet another Hi-C scaffolding tool. *Bioinformatics* **39:** btac808. doi:10.1093/bioinformatics/btac808

Received December 20, 2024; accepted in revised form May 12, 2025.

CSH PRESS GENOME RESEARCH

Verkko2 integrates proximity-ligation data with long-read De Bruijn graphs for efficient telomere-to-telomere genome assembly, phasing, and scaffolding

Dmitry Antipov, Mikko Rautiainen, Sergey Nurk, et al.

Genome Res. 2025 35: 1583-1594 originally published online May 19, 2025
Access the most recent version at doi:10.1101/gr.280383.124