DNA

MDPI

Article

Physlr: Next-Generation Physical Maps

Amirhossein Afshinfard [1,2], Shaun D. Jackman [1], Johnathan Wong [1], Lauren Coombe [1], Justin Chu [1], Vladimir Nikolic [1,2], Gokce Dilek [1], Yaman Malkoç [1], René L. Warren [1] and Inanc Birol [1,3,*]

[1] Canada's Michael Smith Genome Sciences Centre, BC Cancer, Vancouver, BC V5Z 4S6, Canada; aafshinfard@bcgsc.ca (A.A.); sjackman@gmail.com (S.D.J.); jowong@bcgsc.ca (J.W.); lcoombe@bcgsc.ca (L.C.); justinchu1989@gmail.com (J.C.); vnikolic@bcgsc.ca (V.N.); gokcedilek99@gmail.com (G.D.); yamanmalkoc14@gmail.com (Y.M.); rwarren@bcgsc.ca (R.L.W.)

[2] Bioinformatics Graduate Program, University of British Columbia, Vancouver, BC V6T 1Z4, Canada

[3] Department of Medical Genetics, University of British Columbia, Vancouver, BC V6H 3N1, Canada

* Correspondence: ibirol@bcgsc.ca

Abstract: While conventional physical maps helped build most of the reference genomes we use today, generating the maps was prohibitively expensive, and the technology was abandoned in favor of whole-genome shotgun sequencing (WGS). However, genome assemblies generated using WGS data are often less contiguous. We introduce Physlr, a tool that leverages long-range information provided by some WGS technologies to construct next-generation physical maps. These maps have many potential applications in genome assembly and analysis, including, but not limited to, scaffolding. In this study, using experimental linked-read datasets from two humans, we used Physlr to construct chromosome-scale physical maps (NGA50s of 52 Mbp and 70 Mbp). We also demonstrated how these physical maps can help scaffold human genome assemblies generated using various sequencing technologies and assembly tools. Across all experiments, Physlr substantially improved the contiguity of baseline assemblies over state-of-the-art linked-read scaffolders.

Keywords: genome assembly; scaffolding; linked reads; barcode reuse; physical maps; sequencing technologies

Citation: Afshinfard, A.; Jackman, S.D.; Wong, J.; Coombe, L.; Chu, J.; Nikolic, V.; Dilek, G.; Malkoç, Y.; Warren, R.L.; Birol, I. Physlr: Next-Generation Physical Maps. *DNA* **2022**, 2, 116–130. https://doi.org/10.3390/dna2020009

Academic Editor: Darren Griffin

Received: 29 March 2022
Accepted: 7 June 2022
Published: 10 June 2022

1. Introduction

A genome contains genetic instructions for the development, functioning, growth, and reproduction of any known living organism (or virus). Thus, a primary step for many bioinformatics studies is to determine the order of nucleotides in the genome. Despite the rapid technological advances of the past few years, sequencing instruments can generate comparatively short sequence readouts (sequencing reads) redundantly sampled from long target molecules. De novo genome assembly aims to reconstruct the entire genome sequence from overlapping sequencing reads and enable a wide range of downstream studies [1–3].

In the Sanger sequencing era, hierarchical shotgun sequencing was the dominant approach for de novo sequencing and assembly of large genomes (Figure 1a). In this approach, following the initial preparation of large-insert clones, an engineered collection of restriction enzymes was used to cut these molecules into sequence fragments, generating unique fingerprints for each. After measuring these fingerprints by gel electrophoresis, overlaps between molecules were assessed, and a physical map of molecules was created. The resulting map helped distribute the sequencing process and enabled the independent assembly of each molecule. The structure of the physical map then guided the scaffolding of the individually assembled pieces. Due to the use of clones in conventional physical maps, this approach is also known as clone-by-clone or map-based sequencing [4–6].

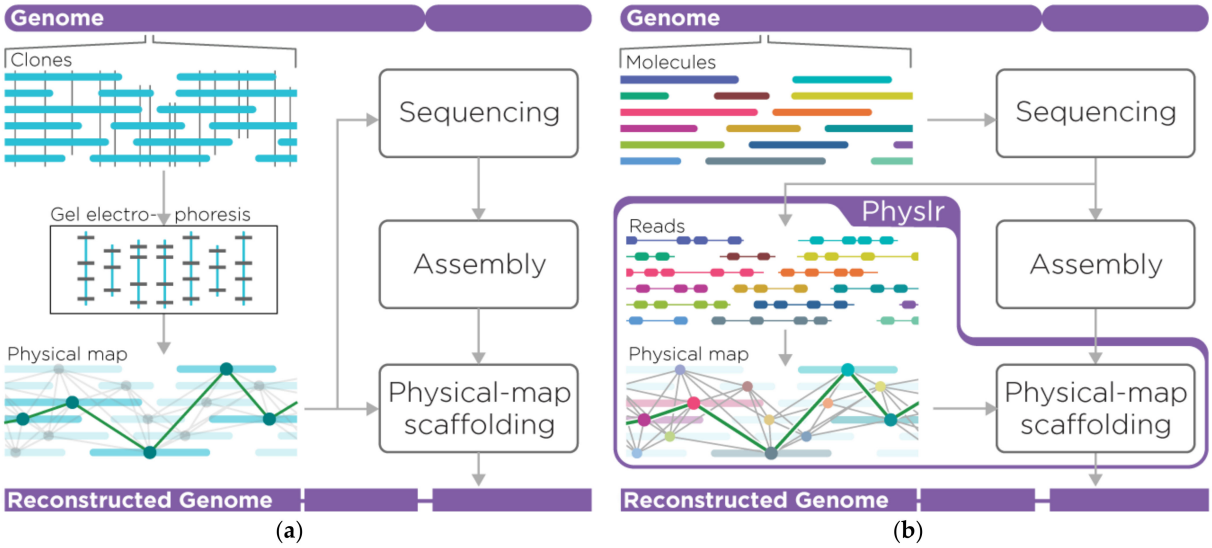

Figure 1. Genome sequencing and assembly paradigms. (**a**) Hierarchical shotgun sequencing uses physical maps to independently sequence and assemble selected clones and scaffold those assemblies to reconstruct the underlying genome. (**b**) Whole-genome shotgun sequencing involves a fast and automated library preparation of DNA fragments (molecules) followed by high-throughput sequencing. Physlr introduces next-generation physical maps to this domain and constructs a map of molecules using reads. The resulting map enables various genomic data analyses, including scaffolding of draft assemblies.

At the turn of the century, this approach facilitated the production of high-quality reference genomes for several model organisms [7–10], enabling a broad range of studies across multiple fields [11,12]. However, this method was labor-intensive, tedious, costly, and time-consuming (13 years for the Human Genome Project [13]). Even in the earlier days of the genomics era, to obtain more affordable human genome assemblies, Venter and colleagues [14] employed whole-genome shotgun sequencing (WGS) [15]. WGS shears the entire genome randomly and combines all the resulting DNA fragments (unordered) for sequencing and assembly (Figure 1b top portion, excluding the Physlr insert). At the crossroad of genome mapping and sequencing technologies, some bioinformatics methodologies combined WGS and the long-range information provided by physical maps to improve the contiguity of WGS genome assemblies at run time [16].

In line with the community's demand for affordable technologies [17], WGS on high-throughput sequencing platforms gradually replaced hierarchical sequencing [18]. While this resulted in rapid growth in de novo sequencing studies, the resulting assemblies were often highly fragmented. To compensate, a diverse range of sequencing technologies have emerged to provide long-range genomics context to genome assemblies, including optical maps [19,20], jumping (mate-pair) sequencing libraries [21], synthetic long reads [22], linked reads [23], long reads [24], and genetic maps [25]. However, not all current pipelines routinely achieve high contiguity [26,27].

This study focuses on leveraging the information content of linked reads for de novo assembly. Linked-read sequencing improves upon short-read sequencing, tagging each short read with a barcode identifier (Supplementary Figure S1). This barcode is identical for all reads originating from the same long molecule (~100 kb), indicating that they come from the same neighborhood in the genome. However, reads from multiple molecules may share the same barcode (barcode reuse) and complicate downstream analyses. Another challenge arises from the fact that molecules are typically sequenced partially (sub-1× coverage). Therefore, the sequence of a single molecule cannot be reconstructed independently. It is feasible, however, to decipher its underlying sequence with redundant molecule sampling.

The error rate of linked reads is much lower than that of long sequencing reads and linked-read molecules tend to span longer regions.

While an early implementation of the linked-read paradigm developed by 10x Genomics (10xG) has been discontinued, more recent developments by MGI (MGIEasy stLFR [28]) and Universal Sequencing (TELL-Seq [29]) continue to offer great potential for de novo sequencing projects. However, we note that barcode reuse remains a challenge for all these platforms, albeit to varying degrees.

On the 31st anniversary of the Human Genome Project [10], we revisited the concept of physical maps employed in that project. We introduce Physlr, a next-generation physical mapping tool using whole-genome sequencing reads (Figure 1b, Physlr insert). Physlr uses the long-range information provided by linked reads to infer a chromosome-scale physical map represented by an overlap graph of the sequenced molecules. This physical map can be used in downstream applications, such as genome assembly/scaffolding, misassembly correction, structural variant detection, and haplotype phasing. Here, we demonstrate how physical maps can be used to generate more contiguous assemblies compared with those produced by the current state-of-the-art linked-read scaffolding tools.

2. Materials and Methods

2.1. Overview of the Pipeline

Physlr runs in two stages: (a) constructing a de novo physical map of DNA fragments (molecules) from which linked reads are generated, and (b) scaffolding a draft genome assembly using the physical map (Figure 2a). It accepts linked reads and a draft assembly (from any sequencing technology) as input to its first and second stage, respectively.

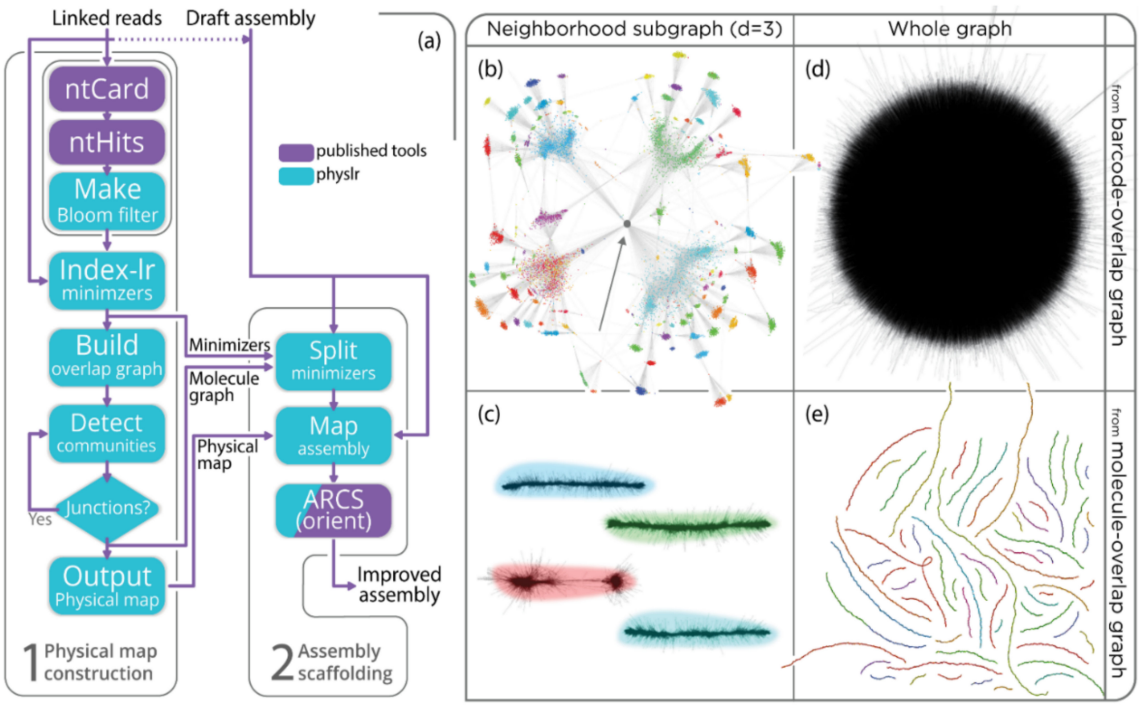

Figure 2. Physlr pipeline. (**a**) Two stages of the Physlr workflow. (**b**–**d**) Physlr subgraphs/graphs before (**b**/**d**) and after (**c**/**e**) deconvoluting molecules (NA24143 stLFR data); colors code the chromosomes to which the elements best map. (**b**) Neighborhood subgraph (distance = 3) of a reused barcode (central grey node, barcode 917_1405_905) connecting four distinct communities, indicating that at least four molecules share this barcode. Other reused barcodes in each community also connect communities from other genomic loci. (**c**) The neighborhood of the same barcode's molecules post-Physlr, deconvoluted into distinct linear graphs. (**d**) Genome-wide barcode overlap graph. (**e**) Long linear graphs after deconvolution (one or a few per chromosome), confirming a chromosome-scale physical map.

In the first stage, Physlr builds a molecule overlap graph. However, linked reads provide information about barcodes rather than molecules. Thus, Physlr first constructs a barcode overlap graph, where a vertex represents a barcode and a weighted edge represents sequence similarity between the reads of the two barcodes (weight being the number of shared minimizers). The topology of communities (sets of vertices densely connected internally and loosely connected externally) around each vertex harbors information about the molecules associated with a given barcode (Figure 2b). We employ a novel algorithm to detect communities and transform the barcode overlap graph into a molecule overlap graph. Next, a representative path, a physical map akin to a golden path used in the Human Genome Project [4,10], is computed and outputted. In the second stage, Physlr uses the physical map to order and orient contigs of the input draft assembly into scaffolds.

2.2. Physlr Implementation, Stage 1: Constructing Physical Maps

In generating physical maps, Physlr transforms linked reads to a barcode graph, then to a molecule graph, and finally to the output physical map (Figure 2a). To mask repetitive sequences efficiently, Physlr first uses ntCard [30] to estimate the cardinality of k-mers (subsequences of length k) of the input data and ntHits [31] to generate a Bloom filter of repetitive k-mers. Subsequently, it generates k-mer minimizer sketches from the reads [32], avoiding minimizers from repeat k-mers found in the ntHits Bloom filter (Figure 2a, index-lr). Accordingly, each barcode is associated with a bag of k-mer minimizers derived from reads of that barcode.

Physlr then constructs a barcode overlap graph wherein vertices represent barcodes and edges connect barcodes sharing a minimum number of minimizers. In an ideal scenario without barcode reuse or repeats, we expect this graph to be composed of multiple connected components, one per chromosome (Figure 2e). We also expect each component to be a long linear graph, or, in graph theory terminology, a graph with a small radius and a large diameter, the former scaling with the average number of molecules covering each base position on the target genome, and the latter scaling with the chromosome length (Figure 2, panels c and e). However, in reality, the graph is more complex due to each set of molecules sharing the same barcode being collapsed into a single vertex (Figure 2, panels b and d).

To transform the barcode graph into a molecule graph, Physlr iterates over all vertices and extracts a neighborhood subgraph for each barcode (Figure 2b). Molecules associated with each barcode originate from independent genomic loci and overlap with different sets of barcodes, thereby creating new communities for the subgraph. Physlr deconvolutes each barcode into molecules using a novel community detection algorithm.

In summary, we randomly split the neighborhood subgraph of a barcode into bins (Supplementary Figure S2). For each bin, we detect biconnected components, and for each component, we calculate the cosine similarity between pairs of vertices. Edges connecting less similar vertices are removed, and connected components of the subgraph are returned as a community. To avoid over-splitting, we compare these communities and merge them if they share enough edges. The algorithm is discussed in further detail in Section 3.

A small fraction of reused barcodes may remain unresolved. Because we expect a small radius and large diameter for molecule overlap graphs, in a given maximum spanning tree (MST) of such graphs, every vertex can possess many short branches and at most two arms (long branches). However, unresolved barcodes convolute the graph, which manifest as vertices with more than two arms in the MST. Physlr employs a linear-time belief propagation algorithm [33] to inform all vertices in the MST on how many arms they possess.

In general, for a vertex-edge pair u and (u, v) in a given tree, this algorithm calculates a belief $b_{\{u,(u,v)\}} = f\left(b_{\{u\}}, b_{\{u,(u,w)\}} \; for\; all\; w \neq u\right)$ where, for a specific problem, $b_{\{u\}}$ is a defined property of the vertex u, and $f()$ is a function. The algorithm starts from leaves in the tree and calculates a belief $b_{\{u,(u,v)\}}$ only when all beliefs required for the calculation are given; then, it passes the belief to vertex v through edge (u, v). As a result, all beliefs are calculated and propagated by passing two messages (beliefs) through each edge of the tree. As the number of edges in a tree is in the same order as the number of vertices, this algorithm runs in linear-time complexity. In our case, $b_{\{u\}} = 1$ and $b_{\{u,(u,v)\}} = b_{\{u\}} + \sum_{all\; w \neq u} b_{\{u,(u,w)\}}$.

After informing vertices about the size of their branches, Physlr flags those with more than two arms as junctions. As they form only a small fraction of barcodes (hundreds out of millions), Physlr reassesses them for community detection with increased sensitivity without heavy computational cost. If unresolved barcodes remain, Physlr removes them from the map. Finally, Physlr calculates an MST of the molecule graph as a representative path (physical map or backbone). This backbone is comprised of ordered lists of molecules, each containing a set of minimizers.

2.3. Physlr Implementation, Stage 2: Scaffolding Draft Assemblies

In the second stage, input draft assemblies are scaffolded using a minimizer-based mapping to the backbone. Because minimizers still associate with barcodes rather than backbone molecules, Physlr assigns barcode minimizers to their associated molecules by selecting the common minimizers shared with neighboring molecules in the molecule graph. We then map the input assembly to the backbone by comparing the minimizers of each sequence to the molecule-assigned minimizers of each molecule.

Finally, we employ ARCS (in ARKS mode) [34,35] to order and orient the input assembly contigs relative to the backbone. ARCS orients scaffold targets by tallying the number of barcodes that share k-mers with both contig ends for each scaffold orientation (head-head, head-tail, tail-head, and tail-tail) and selecting the orientation with the highest number of supporting barcodes. Physlr also extracts distance estimations between scaffold targets calculated by ARCS to report the number of undetermined bases in scaffold gaps.

2.4. Evaluations

We ran Physlr with stLFR sequencing data from two human (*Homo sapiens*) cell lines, NA12878 and NA24143. Reads were downloaded from Genome in a Bottle (Supplementary Table S1) and were reformatted to include barcodes in their headers (standard 10xG linked-read format using BX:Z tag). Physlr provides default parameter settings for various linked-read technologies with the "protocol" option, which controls percent edges (of lower-edge weights) removed from the barcode overlap graph. For the test runs, we set protocol = stlfr (removes 15% of weak overlaps) and used 48 threads.

Physlr builds physical maps de novo but can accept a reference genome to map the minimizers associated with molecules and calculate the length of its physical map in base-pair coordinates for evaluation purposes. This calibration enabled the generation of ideograms (Figure 3) and calculation of NG50 values (maximum length that at least 50% of the target assembly length is in pieces at least this length). In our tests, we used the human genome build GRCh38 (excluding chromosome Y because both cell lines originated from female individuals) as the reference.

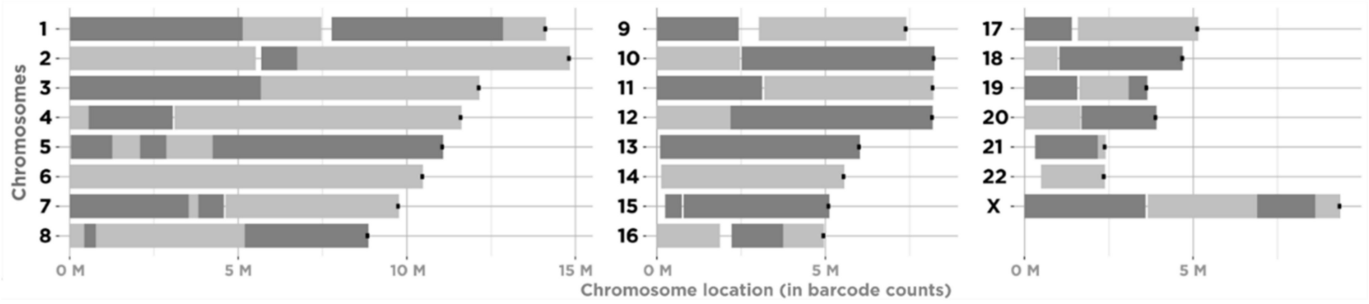

Figure 3. Physlr physical maps for a human genome (NA24143) mapped against the reference. Each reference chromosome is a horizontal line ending in a small black square with a chromosome number on the left. Each grey bar shows the mapping of a backbone piece, with alternating shades marking the start of a new piece. Chromosomes 6, 13, 14, 15, and 22 are spanned by a single backbone, each covering almost the entire chromosome.

We benchmarked Physlr against ARCS (setting the –arks option) and SLR-Superscaffolder (SLR-SS) [36], two state-of-the-art linked-read scaffolding tools. Genomes of the two cell lines (NA12878 and NA24143) were sequenced using four technologies: MGI stLFR, Illumina paired-end and mate-pair (PE+MPET), Oxford Nanopore Technology (ONT), and Pacific Biosciences (PacBio) (Supplementary Table S1). We assembled the stLFR data (same data as in the first stage of Physlr) using Supernova [37] and the PE+MPET data using ABySS 2 [38]. We used Shasta [39] assemblies of the ONT data and Falcon [40] assemblies of the PacBio data, available from Genome in a Bottle (GIAB). We used QUAST [41] to calculate quality metrics for the genome assemblies, from which we visualized the number of misassemblies against the NGA50-NG50 range (NGA50: similar to NG50, but considers alignment blocks instead of contig lengths) as proxies to correctness and contiguity, respectively (showcased in Figure 4). We used GRCh38 as a reference for QUAST; thus, some reported misassemblies were likely genome-specific structural variants for given individuals. More information about assemblies, software versions, and parameters are provided in Supplementary Tables S2–S4.

Genome assembly consistency (Jupiter) plots [42], based on Circos [43], enable a visual evaluation of genome assemblies. The tool plots a draft assembly against a reference assembly on a circle circumference and connects aligned blocks with ribbons. A high-quality assembly results in well-ordered and well-oriented syntenic blocks, while large-scale misassemblies are apparent as crossing ribbons. Figure 5 shows a set of Jupiter plots for the NA12878 assemblies. Here, for each assembly, we calculated the N75 (maximum length that at least 75% of the total assembly length is in pieces at least this length) and L75 (number of scaffolds with length at least N75). Next, for each ternary comparison (each row in Figure 5 comparing baseline, ARCS, and Physlr), we found the minimum of L75s, min-L75, and we plotted only the top min-L75 longest scaffolds for all. For example, Physlr minimized the L75 of all assemblies for the ONT (third row) at 37—hence, the 37 longest scaffolds for all ONT experiments were shown. In other words, we plotted the same number of longest pieces for all assemblies in comparison, and thus a higher proportion of the circle is covered with ribbons for an assembly with higher N75. In the middle of each plot, we also show the percentage of the genome covered by the plotted pieces. As a result, one can quickly compare the assemblies by considering (a) the ribbon coverage, (b) the size of assembly pieces on the right side, and (c) the extent of crossing ribbons (misassemblies) for each circle.

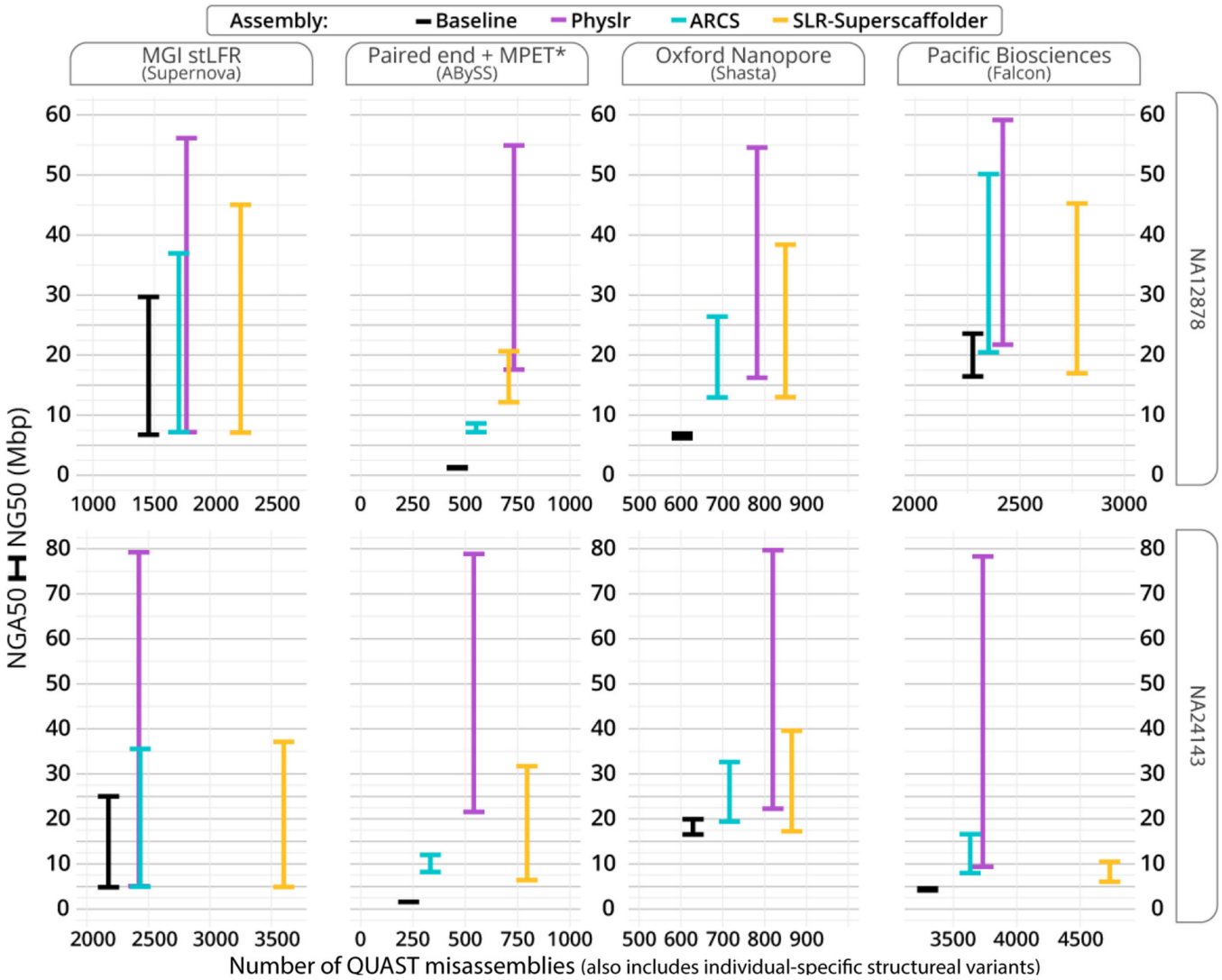

Figure 4. Assembly quality metrics for scaffolding eight human assemblies. Each pair of horizontal lines connected vertically shows a range for NGA50-NG50 of an assembly. Each column corresponds to a sequencing technology (and genome assembly tool, indicated in parentheses) used to generate the baseline assembly, and each row corresponds to a human individual. For each experiment, we evaluated a baseline assembly against scaffolding outputs of Physlr, ARCS [34], and SLR-Superscaffolder [36]. (MPET: Illumina mate-pair).

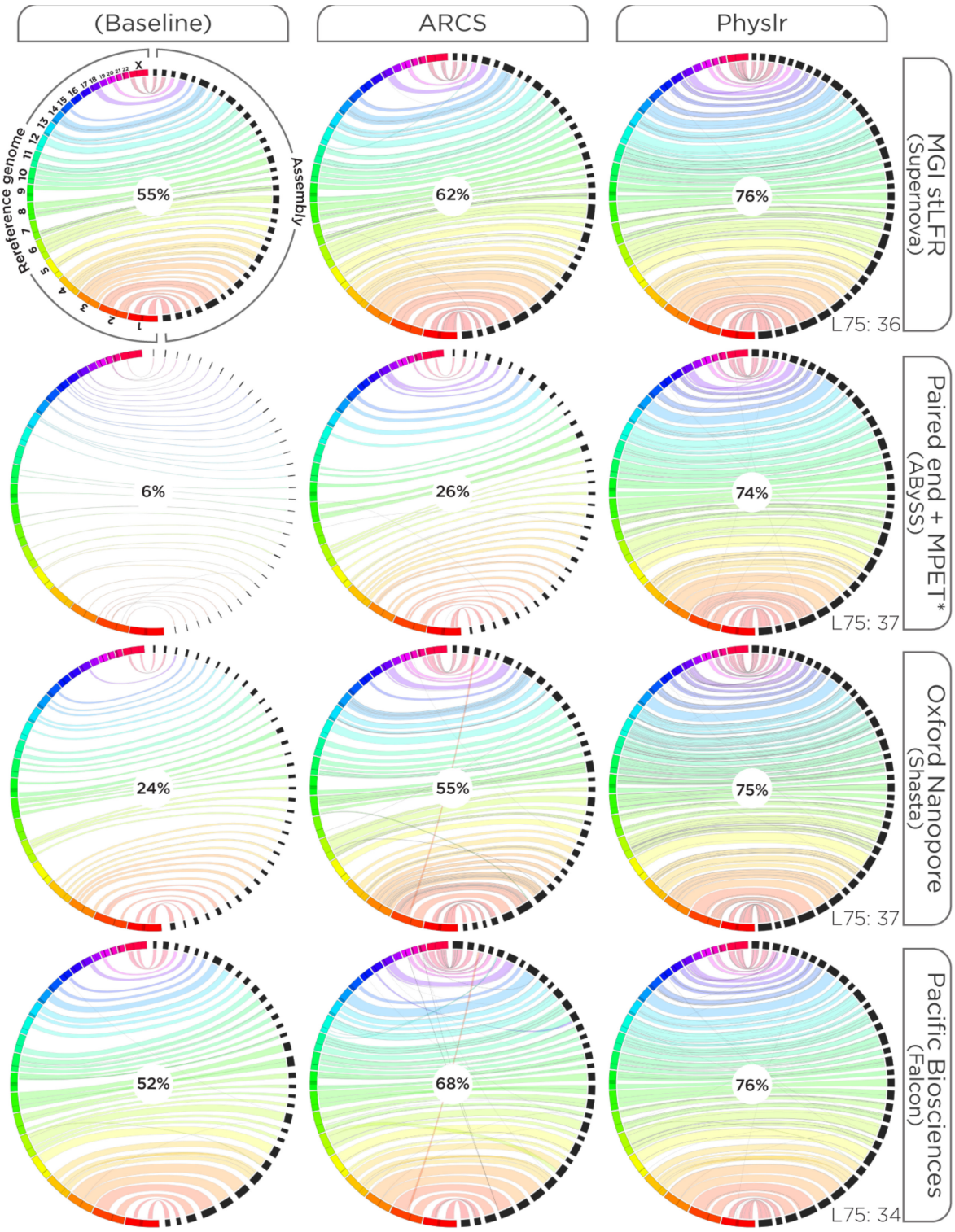

Figure 5. Jupiter plot visualizations for NA12878 assemblies against reference. Each row contains various assemblies (baseline, ARCS [34], and Physlr) for a specific technology and illustrates only a certain number of top largest scaffolds: minimum L75 (min L75) of assemblies in the row (labelled under Physlr, which had the minimum L75 for each row). Physlr consistently presented larger pieces and higher ribbon coverage while keeping crossing ribbons at a low rate. We show ribbon coverage (the percentage of reference covered with min L75 scaffold sequences) in the middle of each plot.

3. Results

3.1. Constructing Physical Maps

We generated physical maps for two human cell lines using stLFR linked reads (Section 2.4. Evaluations). A physical map of molecules comprises multiple connected components—one per chromosome in the best-case scenario. In assessing the contiguity of the maps, we found the minimum number of graph components that contained at least 75% of the vertices (molecules) were 32 and 44 for NA12878 and NA24143, respectively. This suggests that, at this length cutoff, Physlr produced nearly two components per chromosome, on average. Next, we converted the coordinates to base pairs and calculated the NGA50 values of the maps as 52.4 Mbp and 70.49 Mbp for NA12878 and NA24143, respectively.

As seen in the ideograms generated from the NA24143 physical map (Figure 3), five chromosomes were covered by a single piece, with the physical maps spanning over centromeres for chromosomes 6 and 19. Nine other chromosomes were covered with two backbone pieces each (one backbone per chromosome arm). The most fragmented chromosome was chromosome 5, covered with five backbone pieces. Similar results were observed for the physical map for NA12878, as shown in Supplementary Figure S3.

To demonstrate that Physlr is robust to changes in the linked-read technology, we also built physical maps using 10xG Chromium data for NA24143. The resulting physical map had an estimated NGA50 of over 70 Mbp (Supplementary Results).

3.2. Scaffolding Draft Assemblies

We evaluated the potential value of physical maps generated by Physlr in scaffolding draft assemblies (Section 2.4. Evaluations). Physlr increased the NG50 and NGA50 across all experiments (Figure 4). For example, it improved the NG50 and NGA50 values of two short-read assemblies (PE+MPET) by over 45-fold and 12-fold, respectively, and increased the number of misassemblies by less than 1-fold on average. In another example, Physlr improved a long-read (ONT) assembly of NA12878: 7.9/2.8 -fold change (54.4/17.3 Mbp) in NG/NGA50 with less than 0.3-fold increase in misassemblies. For the same experiment, Physlr outperformed ARCS with a 2.1/1.2-fold change in NG/NGA50 values.

Overall, Physlr-scaffolded assemblies reached NGA50s of up to 21.7 Mbp and 22.3 Mbp for NA12878 and NA24143, respectively. The NG50 improved consistently for all Physlr assemblies and ranged between 54–60 Mbp, and 78–80 Mbp, respectively. However, the NG50s of ARCS and SLR-SS scaffolded assemblies varied in wider ranges for different technologies. While Physlr improved the contiguity of assemblies over other tools, the number of misassemblies was either lower or marginally higher in respect to the substantial increase in the contiguity. All tools only slightly enhanced the NGA50 of the Supernova baseline assemblies; these baseline assemblies contained more errors compared to other baseline assemblies.

To better visualize the metrics in Figure 4 for the NA12878 assemblies, we generated Jupiter plots (Figure 5). Each row shows a baseline assembly, its scaffolding with Physlr, and its scaffolding with ARCS. For all rows, Physlr achieved the lowest L75s, maximized the ribbon coverage, contributed larger contigs, and produced only a few inconsistent ribbons, all of which suggest a better performance as discussed above (Section 2.4. Evaluations).

We also successfully scaffolded the same set of baseline assemblies for NA24143 using a physical map of 10x Genomics linked reads and presented outcomes in Supplementary Results (Physlr for 10xG Chromium data) and Supplementary Table S6.

3.3. Deconvoluting Barcodes via Community Detection

Barcode reuse is a fundamental challenge for linked-read technologies. While all other reference-free tools ignore barcode reuse (except for Minerva [44], only applicable for metagenomics), Physlr uses long-range information aggregated in the graph to split barcodes into molecules. Each vertex b_i in the barcode graph comprises multiple hidden molecules $mol_{i,1}$ to $mol_{i,M}$ (Supplementary Figure S4). Each constituent molecule $mol_{i,m}$

originates from a different region in the genome. Thus, each molecule tends to overlap with a different set of molecules, all of which originate from the same genomic site. Consequently, $mol_{i,m}$ connects vertex b_i to barcodes $C_{i,m}$, a set (community) of barcodes that are highly connected internally because they contain molecules from the same genomic region. In other words, $C_{i,m}$ is a strong community of barcodes adjacent to b_i through $mol_{i,1}$. Ultimately, b_i is connected to multiple communities $C_{i,1}$ to $C_{i,M}$, which are strongly connected internally and weakly connected to one another, as each tends to originate from a different region. Thus, we can deconvolute a barcode-vertex b_i by detecting community patterns ($C_{i,1}$ to $C_{i,M}$) in the neighborhood subgraph. Figure 2b illustrates one real-world example of a barcode's neighborhood in presence of barcode reuse; four distinct communities imply the barcode contains at least four molecules.

Following this logic, Physlr iterates over all vertices and deconvolutes each barcode into its constituent molecules one at a time (Supplementary Figure S5). To achieve this deconvolution, we inspect each barcode's neighborhood subgraph, the vertex-induced subgraph of a barcode's immediate neighbors. We expect this subgraph to contain multiple communities, one per molecule (Figure 2b). Communities are detected (explained below), and the focal barcode is split into multiple molecule vertices, one per community; we connect each molecule-vertex to all vertices in its relative community.

Physlr detects communities in millions of subgraphs, each of which comprises hundreds to thousands of vertices. Although community detection is a well-studied topic, current state-of-the-art algorithms [45,46] failed to scale up for Physlr subgraphs (Supplementary Table S8). Hence, we devised a novel algorithm for community detection.

First, we randomly split larger subgraphs into smaller (sub-)subgraphs (Supplementary Figure S2). Next, we detect biconnected components. For each component, we connect every vertex to its second-order neighbors to increase communities' interconnectivity, increasing the signal-to-noise ratio. This is implemented by squaring the adjacency matrix. We then calculate a 2-dimensional cosine similarity matrix CS of the adjacency matrix (implemented in the same manner). The value of each element $CS_{i,j}$ reflects the extent of shared neighbors between nodes i and j. We adopt a threshold to remove weak connections and report connected components as subcommunities. Finally, we merge the resulting subcommunities if the merging increases the modularity (measures the strength of division of a network into modules) [47].

In summary, we use a divide and conquer approach: we divide the subgraphs, detect the communities in each division, and re-join related communities. As a result, despite processing millions of subgraphs, Physlr runs comparatively fast (Supplementary Figure S6 and Supplementary Tables S7 and S8).

Physlr also implements a divide-and-conquer version of other community detection algorithms, including tri-connected components, k-clique percolation [45], and Louvain community detection [46], and allows for a customized combination of all these choices in an iterative manner. A performance comparison between some potential combinations is provided in the Supplementary Results (Supplementary Table S8).

4. Discussion

Sequencing technologies are rapidly evolving, providing longer-range information. However, it is not yet a routine task to achieve assemblies with contiguity comparable to studies that benefit from the long-range information inherent in conventional physical maps. We revived the concept and presented a tool, Physlr, that constructs next-generation physical maps based on linked-read data. We showed that Physlr maps can almost cover human chromosomes in 1–5 pieces (<2.5 pieces on average). Furthermore, we used these generated physical maps to scaffold various draft human genomes assembled using four different sequencing platforms, including short-, linked- and long-read technologies, and showed substantial contiguity gains in each scenario. This suggests that Physlr can substantially boost assembly projects with linked-read data. Physlr may also be used to improve recently published linked-read genome assemblies [48–52] reusing the same data.

While traditional maps were mainly used in genome assembly and scaffolding projects (due to high mapping cost), their modern alternatives such as optical maps, long reads and, more recently, linked reads are used in a broader range of applications: personal genome assembly [53], structural variation and recombination detection [54–57], haplotyping assemblies and variants [23,58,59], assembly correction and evaluation [60,61], etc. In the same manner, next-generation physical maps would be applicable to this analysis spectrum. Physlr compiles long-range information from separated molecules into a unified physical map, enabling "longer-range" more robust inference—as demonstrated for scaffolding. Due to the high contiguity of Physlr physical maps, they have great potential to provide a big picture of the genome structure and to serve various downstream genomic studies.

We based our work on linked reads since their molecules tend to span longer genomic loci than long reads, and closer to molecules in conventional physical maps. Linked reads are sequenced through short-read sequencing instruments and thus have a very low error rate, which enables decent overlap detection in Physlr. As long-read technologies gradually close in on short linked reads in terms of error-rate, cost, and especially molecule (fragment) size, Physlr may be adapted to build a physical map of long reads.

The contiguity proxy NG50 can increase by solely introducing new misassemblies. To validate that Physlr increases the contiguity, we also looked at NGA50 which considers alignment blocks instead of contigs lengths and thus rules in the effect of misassemblies. Additionally, Figure 5 visually confirms that the contiguity of Physlr scaffolds was not due to large-scale misassemblies. Moreover, only a fraction of the joins that Physlr made were flagged as QUAST misassemblies (some of which are likely individual-specific structural variants).

Physlr currently trusts the input assembly over its physical map where they contradict. Thus, an initial assembly containing numerous misassemblies may restrain physical maps' potential by preventing Physlr from correcting the assembly and making many potential joins; in two of our experiments, Supernova generated numerous misassemblies in baseline assemblies and Physlr (and other tools) increased the contiguity only slightly. Thus, in devising an assembly pipeline, we suggest including an assembly correction tool like Tigmint [61] prior to Physlr scaffolding. Sequencing technology-specific assemblers are available to use upstream in the pipeline, as needed [37–39,62–64].

Physlr requires at least two physical map nodes (molecules) mapping to a contig of the input assembly to anchor, orient, and scaffold the contig. Thus, we recommend that the input assembly contains contigs larger than the size of one molecule (>100 kbp).

To the best of our knowledge, Physlr is the only scalable tool that can deconvolute reused barcodes into their associated molecules de novo. For this purpose, we devised a novel community detection algorithm based on a divide-and-conquer approach, cosine similarity, and k-clique percolation, while its customizable pipeline works with other well-known algorithms as well. The community detection algorithms' performance heavily relies on the topological nature of the network/graph [65]. Our algorithm outperformed all others on our overlap graphs; thus, we promote it as a potential community detection algorithm that would suit other studies.

5. Conclusions

With Physlr, we introduced next-generation physical maps based on linked reads and demonstrated the potential of the physical maps to benefit genomic studies by showcasing improvements in scaffolding genome assemblies. Physlr outperformed state-of-the-art linked-read scaffolders and substantially increased the contiguity (both NG50 and NGA50, for all eight human assemblies considered) while performing well in keeping misassemblies comparatively low.

Physlr may be used to scaffold genome assemblies in linked-read or hybrid projects or to generate physical maps and empower other downstream applications and studies.

Physlr is an open-source project and publicly available under GNU General Public License v3.0 license at https://github.com/bcgsc/physlr (accessed on 6 June 2022).

Supplementary Materials: The following are available online at https://www.mdpi.com/article/10.3390/dna2020009/s1, Supplementary Results including (a) Evaluations and (b) Physlr for 10xG Chromium data, Supplementary Table S1: H. sapiens stLFR linked-read datasets used for test runs, Supplementary Table S2: Baseline H. sapiens assemblies used for scaffolding, Supplementary Table S3: Baseline assembly quality statistics (assessed using Quast), Supplementary Table S4: Various tools used for scaffolding or for evaluation, Supplementary Table S5: Quality statistics for scaffolded assemblies (using Quast), Supplementary Table S6: Scaffolding baseline assemblies with Physlr using 10xG Chromium physical maps (assessed using Quast), Supplementary Table S7: Resource profiling of all tools, Supplementary Table S8: Profiling of community detection algorithms on 10% of the subgraphs, Supplementary Figure S1: Linked-read sequencing data and barcode overlap graphs, Supplementary Figure S2: Physlr's community detection algorithm for a subgraph, Supplementary Figure S3: Physlr physical maps for a human genome (NA24143) mapped against the reference, Supplementary Figure S4: Barcode re-use and the barcode overlap graph, Supplementary Figure S5: Barcode to molecule deconvolution, Supplementary Figure S6: Run-time and peak memory usage.

Author Contributions: A.A., S.D.J. and I.B. designed the method and algorithms. A.A., S.D.J., J.W., L.C., V.N. and J.C. implemented the Physlr software. G.D. and Y.M. tested Physlr. A.A. and J.W. conducted the benchmarking experiments. A.A., J.W., L.C., R.L.W. and I.B. analyzed the results. A.A. drafted the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Genome BC and Genome Canada [243FOR, 281ANV]; and the National Institutes of Health [2R01HG007182-04A1]. A.A.'s work was supported by a Ph.D. fellowship (4YF) from the University of British Columbia. The content of this article is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health or other funding organizations.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All the necessary data to regenerate this study, including the experiments, are available as listed in the Supplementary Materials.

Acknowledgments: We appreciate Ka Ming Nip's support and contribution through discussions during Physlr's development.

Conflicts of Interest: S.D.J. started working for 10x Genomics after the completion of this project. The other authors declare no conflict of interest. Neither 10x Genomics nor the funders had any role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Lewin, H.A.; Larkin, D.M.; Pontius, J.; O'Brien, S.J. Every Genome Sequence Needs a Good Map. *Genome Res.* **2009**, *19*, 1925. [CrossRef] [PubMed]
2. Rice, E.S.; Green, R.E. New Approaches for Genome Assembly and Scaffolding. *Annu. Rev. Anim. Biosci.* **2019**, *7*, 17–40. [CrossRef] [PubMed]
3. Giani, A.M.; Gallo, G.R.; Gianfranceschi, L.; Formenti, G. Long Walk to Genomics: History and Current Approaches to Genome Sequencing and Assembly. *Comput. Struct. Biotechnol. J.* **2020**, *18*, 9–19. [CrossRef] [PubMed]
4. McPherson, J.D.; Marra, M.; Hillier, L.D.; Waterston, R.H.; Chinwalla, A.; Wallis, J.; Sekhon, M.; Wylie, K.; Mardis, E.R.; Wilson, R.K.; et al. A Physical Map of the Human Genome. *Nature* **2001**, *409*, 934–941. [CrossRef]
5. Zhang, H.B.; Wu, C. BAC as Tools for Genome Sequencing. *Plant Physiol. Biochem.* **2001**, *39*, 195–209. [CrossRef]
6. Green, E.D. Strategies for the Systematic Sequencing of Complex Genomes. *Nat. Rev. Genet.* **2001**, *2*, 573–583. [CrossRef]
7. Goffeau, A.; Aert, R.; Agostini-Carbone, M.L.; Ahmed, A.; Aigle, M.; Alberghina, L.; Albermann, K.; Albers, M.; Aldea, M.; Alexandraki, D.; et al. The Yeast Genome Directory. *Nature* **1997**, *387*, 5. [CrossRef]
8. Equence, C.E.S.; Iology, T.O.B.; The, C.; Consortium, S. Genome Sequence of the Nematode C. Elegans: A Platform for Investigating Biology. *Science* **1998**, *282*, 2012–2018. [CrossRef]
9. Mayer, K.; Schüller, C.; Wambutt, R.; Murphy, G.; Volckaert, G.; Pohl, T.; Düsterhöft, A.; Stiekema, W.; Entian, K.D.; Terryn, N.; et al. Sequence and Analysis of Chromosome 4 of the Plant Arabidopsis Thaliana. *Nature* **1999**, *402*, 769–777. [CrossRef]
10. Lander, E.S.; Linton, L.M.; Birren, B.; Nusbaum, C.; Zody, M.C.; Baldwin, J.; Devon, K.; Dewar, K.; Doyle, M.; Fitzhugh, W.; et al. Initial Sequencing and Analysis of the Human Genome. *Nature* **2001**, *409*, 860–921. [CrossRef]

11. Collins, F.S.; McKusick, V.A. Implications of the Human Genome Project for Medical Science. *J. Am. Med. Assoc.* **2001**, *285*, 540–544. [CrossRef] [PubMed]
12. Skolnick, J.; Fetrow, J.S. From Genes to Protein Structure and Function: Novel Applications of Computational Approaches in the Genomic Era. *Trends Biotechnol.* **2000**, *18*, 34–39. [CrossRef]
13. Human Genome Project FAQ. Available online: https://www.genome.gov/human-genome-project/Completion-FAQ (accessed on 16 October 2021).
14. Craig Venter, J.; Adams, M.D.; Myers, E.W.; Li, P.W.; Mural, R.J.; Sutton, G.G.; Smith, H.O.; Yandell, M.; Evans, C.A.; Holt, R.A.; et al. The Sequence of the Human Genome. *Science* **2001**, *291*, 1304–1351. [CrossRef]
15. Weber, J.L.; Myers, E.W. Human Whole-Genome Shotgun Sequencing. *Genome Res.* **1997**, *7*, 401–409. [CrossRef] [PubMed]
16. Warren, R.L.; Varabei, D.; Platt, D.; Huang, X.; Messina, D.; Yang, S.P.; Kronstad, J.W.; Krzywinski, M.; Warren, W.C.; Wallis, J.W.; et al. Physical Map-Assisted Whole-Genome Shotgun Sequence Assemblies. *Genome Res.* **2006**, *16*, 768. [CrossRef] [PubMed]
17. Schloss, J.A. How to Get Genomes at One Ten-Thousandth the Cost. *Nat. Biotechnol.* **2008**, *26*, 1113–1115. [CrossRef]
18. Reuter, J.A.; Spacek, D.V.; Snyder, M.P. Review High-Throughput Sequencing Technologies. *Mol. Cell* **2015**, *58*, 586–597. [CrossRef]
19. Das, S.K.; Austin, M.D.; Akana, M.C.; Deshpande, P.; Cao, H.; Xiao, M. Single Molecule Linear Analysis of DNA in Nano-Channel Labeled with Sequence Specific Fluorescent Probes. *Nucleic Acids Res.* **2010**, *38*, e177. [CrossRef]
20. Lam, E.T.; Hastie, A.; Lin, C.; Ehrlich, D.; Das, S.K.; Austin, M.D.; Deshpande, P.; Cao, H.; Nagarajan, N.; Xiao, M.; et al. Genome Mapping on Nanochannel Arrays for Structural Variation Analysis and Sequence Assembly. *Nat. Biotechnol.* **2012**, *30*, 771–776. [CrossRef]
21. Williams, L.J.S.; Tabbaa, D.G.; Li, N.; Berlin, A.M.; Shea, T.P.; MacCallum, I.; Lawrence, M.S.; Drier, Y.; Getz, G.; Young, S.K.; et al. Paired-End Sequencing of Fosmid Libraries by Illumina. *Genome Res.* **2012**, *22*, 2241. [CrossRef]
22. Li, R.; Hsieh, C.L.; Young, A.; Zhang, Z.; Ren, X.; Zhao, Z. Illumina Synthetic Long Read Sequencing Allows Recovery of Missing Sequences Even in the "Finished" C. Elegans Genome. *Sci. Rep.* **2015**, *5*, 10814. [CrossRef] [PubMed]
23. Zheng, G.X.Y.; Lau, B.T.; Schnall-Levin, M.; Jarosz, M.; Bell, J.M.; Hindson, C.M.; Kyriazopoulou-Panagiotopoulou, S.; Masquelier, D.A.; Merrill, L.; Terry, J.M.; et al. Haplotyping Germline and Cancer Genomes with High-Throughput Linked-Read Sequencing. *Nat. Biotechnol.* **2016**, *34*, 303–311. [CrossRef]
24. Pollard, M.O.; Gurdasani, D.; Mentzer, A.J.; Porter, T.; Sandhu, M.S. Long Reads: Their Purpose and Place. *Hum. Mol. Genet.* **2018**, *27*, R234–R241. [CrossRef] [PubMed]
25. Kai, W.; Kikuchi, K.; Tohari, S.; Chew, A.K.; Tay, A.; Fujiwara, A.; Hosoya, S.; Suetake, H.; Naruse, K.; Brenner, S.; et al. Integration of the Genetic Map and Genome Assembly of Fugu Facilitates Insights into Distinct Features of Genome Evolution in Teleosts and Mammals. *Genome Biol. Evol.* **2011**, *3*, 424–442. [CrossRef] [PubMed]
26. Di Genova, A.; Buena-Atienza, E.; Ossowski, S.; Sagot, M.-F. Efficient Hybrid de Novo Assembly of Human Genomes with WENGAN. *Nat. Biotechnol.* **2020**, *39*, 422–430. [CrossRef]
27. Nurk, S.; Koren, S.; Rhie, A.; Rautiainen, M.; Bzikadze, A.V.; Mikheenko, A.; Vollger, M.R.; Altemose, N.; Uralsky, L.; Gershman, A.; et al. The Complete Sequence of a Human Genome. *Science* **2022**, *376*, 44–53. [CrossRef]
28. Wang, O.; Chin, R.; Cheng, X.; Yan Wu, M.K.; Mao, Q.; Tang, J.; Sun, Y.; Anderson, E.; Lam, H.K.; Chen, D.; et al. Efficient and Unique Cobarcoding of Second-Generation Sequencing Reads from Long DNA Molecules Enabling Cost-Effective and Accurate Sequencing, Haplotyping, and de Novo Assembly. *Genome Res.* **2019**, *29*, 798–808. [CrossRef]
29. Chen, Z.; Pham, L.; Wu, T.C.; Mo, G.; Xia, Y.; Chan, P.L.; Porter, D.; Phan, T.; Che, H.; Tran, H.; et al. Ultralow-Input Single-Tube Linked-Read Library Method Enables Short-Read Second-Generation Sequencing Systems to Routinely Generate Highly Accurate and Economical Long-Range Sequencing Information. *Genome Res.* **2020**, *30*, 898–909. [CrossRef]
30. Mohamadi, H.; Khan, H.; Birol, I. NtCard: A Streaming Algorithm for Cardinality Estimation in Genomics Data. *Bioinformatics* **2017**, *33*, 1324–1330. [CrossRef]
31. Mohamadi, H.; Chu, J.; Coombe, L.; Warren, R.; Birol, I. NtHits: De Novo Repeat Identification of Genomics Data Using a Streaming Approach. *bioRxiv* **2020**. [CrossRef]
32. Roberts, M.; Hayes, W.; Hunt, B.R.; Mount, S.M.; Yorke, J.A. Reducing Storage Requirements for Biological Sequence Comparison. *Bioinformatics* **2004**, *20*, 3363–3369. [CrossRef] [PubMed]
33. Pearl, J. Reverend Bayes on Inference Engines: A Distributed Hierarchical Approach. In Proceedings of the Second AAAI Conference on Artificial Intelligence, Pittsburgh, PA, USA, 18 August 1982; pp. 133–136.
34. Coombe, L.; Zhang, J.; Vandervalk, B.P.; Chu, J.; Jackman, S.D.; Birol, I.; Warren, R.L. ARKS: Chromosome-Scale Scaffolding of Human Genome Drafts with Linked Read Kmers. *BMC Bioinform.* **2018**, *19*, 234. [CrossRef] [PubMed]
35. Yeo, S.; Coombe, L.; Warren, R.L.; Chu, J.; Birol, I. ARCS: Scaffolding Genome Drafts with Linked Reads. *Bioinformatics* **2018**, *34*, 725–731. [CrossRef] [PubMed]
36. Guo, L.; Xu, M.; Wang, W.; Gu, S.; Zhao, X.; Chen, F.; Wang, O.; Xu, X.; Seim, I.; Fan, G.; et al. SLR-Superscaffolder: A de Novo Scaffolding Tool for Synthetic Long Reads Using a Top-to-Bottom Scheme. *BMC Bioinform.* **2021**, *22*, 158. [CrossRef] [PubMed]
37. Weisenfeld, N.I.; Kumar, V.; Shah, P.; Church, D.M.; Jaffe, D.B. Direct Determination of Diploid Genome Sequences. *Genome Res.* **2017**, *27*, 757–767. [CrossRef]
38. Jackman, S.D.; Vandervalk, B.P.; Mohamadi, H.; Chu, J.; Yeo, S.; Hammond, S.A.; Jahesh, G.; Khan, H.; Coombe, L.; Warren, R.L.; et al. ABySS 2.0: Resource-Efficient Assembly of Large Genomes Using a Bloom Filter Effect of Bloom Filter False Positive Rate. *Genome Res.* **2017**, *27*, 768–777. [CrossRef]

39. Shafin, K.; Pesout, T.; Lorig-Roach, R.; Haukness, M.; Olsen, H.E.; Bosworth, C.; Armstrong, J.; Tigyi, K.; Maurer, N.; Koren, S.; et al. Nanopore Sequencing and the Shasta Toolkit Enable Efficient de Novo Assembly of Eleven Human Genomes. *Nat. Biotechnol.* **2020**, *38*, 1044–1053. [CrossRef]
40. Chin, C.-S.; Peluso, P.; Sedlazeck, F.J.; Nattestad, M.; Concepcion, G.T.; Clum, A.; Dunn, C.; O'Malley, R.; Figueroa-Balderas, R.; Morales-Cruz, A.; et al. Phased Diploid Genome Assembly with Single-Molecule Real-Time Sequencing. *Nat. Methods* **2016**, *13*, 1050–1054. [CrossRef]
41. Gurevich, A.; Saveliev, V.; Vyahhi, N.; Tesler, G. QUAST: Quality Assessment Tool for Genome Assemblies. *Bioinformatics* **2013**, *29*, 1072–1075. [CrossRef]
42. Chu, J. Jupiter Plot: A Circos-Based Tool to Visualize Genome Assembly Consistency (Version 1.0). Zenodo. 2018. Available online: https://zenodo.org/record/1241235#.YqEDN6hlBD9 (accessed on 6 June 2022).
43. Krzywinski, M.; Schein, J.; Birol, I.; Connors, J.; Gascoyne, R.; Horsman, D.; Jones, S.J.; Marra, M.A. Circos: An Information Aesthetic for Comparative Genomics. *Genome Res.* **2009**, *19*, 1639–1645. [CrossRef]
44. Danko, D.C.; Meleshko, D.; Bezdan, D.; Mason, C.; Hajirasouliha, I. Minerva: An Alignment- and Reference-Free Approach to Deconvolve Linked-Reads for Metagenomics. *Genome Res.* **2019**, *29*, 116–124. [CrossRef] [PubMed]
45. Palla, G.; Derényi, I.; Farkas, I.; Vicsek, T. Uncovering the Overlapping Community Structure of Complex Networks in Nature and Society. *Nature* **2005**, *435*, 814–818. [CrossRef] [PubMed]
46. Blondel, V.D.; Guillaume, J.L.; Lambiotte, R.; Lefebvre, E. Fast Unfolding of Communities in Large Networks. *J. Stat. Mech. Theory Exp.* **2008**, *2008*, P10008. [CrossRef]
47. Newman, M.E.J.; Girvan, M. Finding and Evaluating Community Structure in Networks. *Phys. Rev. E* **2004**, *69*, 026113. [CrossRef] [PubMed]
48. Mori, B.A.; Coutu, C.; Chen, Y.H.; Campbell, E.O.; Dupuis, J.R.; Erlandson, M.A.; Hegedus, D.D. De Novo Whole Genome Assembly of the Swede Midge (*Contarinia nasturtii*), a Specialist of Brassicaceae, Using Linked-Read Sequencing. *Genome Biol. Evol.* **2021**, *13*, evab036. [CrossRef]
49. Engler, J.O.; Lawrie, Y.; Gansemans, Y.; van Nieuwerburgh, F.; Suh, A.; Lens, L. Genome Report: De Novo Genome Assembly and Annotation for the Taita White-Eye (Zosterops Silvanus). *bioRxiv* **2020**. [CrossRef]
50. Brian Simison, W.; Parham, J.F.; Papenfuss, T.J.; Lam, A.W.; Henderson, J.B. An Annotated Chromosome-Level Reference Genome of the Red-Eared Slider Turtle (Trachemys Scripta Elegans). *Genome Biol. Evol.* **2020**, *12*, 456–462. [CrossRef]
51. Roodgar, M.; Babveyh, A.; Nguyen, L.H.; Zhou, W.; Sinha, R.; Lee, H.; Hanks, J.B.; Avula, M.; Jiang, L.; Jian, R.; et al. Chromosome-Level de Novo Assembly of the Pig-Tailed Macaque Genome Using Linked-Read Sequencing and HiC Proximity Scaffolding. *Gigascience* **2020**, *9*, giaa069. [CrossRef]
52. Helmkampf, M.; Bellinger, M.R.; Geib, S.M.; Sim, S.B.; Takabayashi, M. Draft Genome of the Rice Coral Montipora Capitata Obtained from Linked-Read Sequencing. *Genome Biol. Evol.* **2019**, *11*, 2045–2054. [CrossRef]
53. Zhou, X.; Zhang, L.; Weng, Z.; Dill, D.L.; Sidow, A. Aquila Enables Reference-Assisted Diploid Personal Genome Assembly and Comprehensive Variant Detection Based on Linked Reads. *Nat. Commun.* **2021**, *12*, 1077. [CrossRef]
54. Onore, M.E.; Torella, A.; Musacchia, F.; D'Ambrosio, P.; Zanobio, M.; del Vecchio Blanco, F.; Piluso, G.; Nigro, V. Linked-Read Whole Genome Sequencing Solves a Double DMD Gene Rearrangement. *Genes* **2021**, *12*, 133. [CrossRef] [PubMed]
55. Fang, L.; Kao, C.; Gonzalez, M.V.; Mafra, F.A.; Pellegrino da Silva, R.; Li, M.; Wenzel, S.S.; Wimmer, K.; Hakonarson, H.; Wang, K. LinkedSV for Detection of Mosaic Structural Variants from Linked-Read Exome and Genome Sequencing Data. *Nat. Commun.* **2019**, *10*, 5585. [CrossRef] [PubMed]
56. Teague, B.; Waterman, M.S.; Goldstein, S.; Potamousis, K.; Zhou, S.; Reslewic, S.; Sarkar, D.; Valouev, A.; Churas, C.; Kidd, J.M.; et al. High-Resolution Human Genome Structure by Single-Molecule Analysis. *Proc. Natl. Acad. Sci. USA* **2010**, *107*, 10848–10853. [CrossRef] [PubMed]
57. Dréau, A.; Venu, V.; Avdievich, E.; Gaspar, L.; Jones, F.C. Genome-Wide Recombination Map Construction from Single Individuals Using Linked-Read Sequencing. *Nat. Commun.* **2019**, *10*, 4309. [CrossRef]
58. Xu, M.; Guo, L.; Du, X.; Li, L.; Peters, B.A.; Deng, L.; Wang, O.; Chen, F.; Wang, J.; Jiang, Z.; et al. Accurate Haplotype-Resolved Assembly Reveals the Origin of Structural Variants for Human Trios. *Bioinformatics* **2021**, *37*, 2095–2102. [CrossRef]
59. Chaisson, M.J.P.; Sanders, A.D.; Zhao, X.; Malhotra, A.; Porubsky, D.; Rausch, T.; Gardner, E.J.; Rodriguez, O.L.; Guo, L.; Collins, R.L.; et al. Multi-Platform Discovery of Haplotype-Resolved Structural Variation in Human Genomes. *Nat. Commun.* **2019**, *10*, 1784. [CrossRef]
60. Udall, J.A.; Dawe, R.K. Is It Ordered Correctly? Validating Genome Assemblies by Optical Mapping. *Plant Cell* **2018**, *30*, 7–14. [CrossRef]
61. Jackman, S.D.; Coombe, L.; Chu, J.; Warren, R.L.; Vandervalk, B.P.; Yeo, S.; Xue, Z.; Mohamadi, H.; Bohlmann, J.; Jones, S.J.M.; et al. Tigmint: Correcting Assembly Errors Using Linked Reads from Large Molecules. *BMC Bioinform.* **2018**, *19*, 393. [CrossRef]
62. Rhie, A.; McCarthy, S.A.; Fedrigo, O.; Damas, J.; Formenti, G.; Koren, S.; Uliano-Silva, M.; Chow, W.; Fungtammasan, A.; Kim, J.; et al. Towards Complete and Error-Free Genome Assemblies of All Vertebrate Species. *Nature* **2021**, *592*, 737–746. [CrossRef]
63. Cheng, H.; Concepcion, G.T.; Feng, X.; Zhang, H.; Li, H. Haplotype-Resolved de Novo Assembly Using Phased Assembly Graphs with Hifiasm. *Nat. Methods* **2021**, *18*, 170–175. [CrossRef]

64. Nurk, S.; Walenz, B.P.; Rhie, A.; Vollger, M.R.; Logsdon, G.A.; Grothe, R.; Miga, K.H.; Eichler, E.E.; Phillippy, A.M.; Koren, S. HiCanu: Accurate Assembly of Segmental Duplications, Satellites, and Allelic Variants from High-Fidelity Long Reads. *Genome Res.* **2020**, *30*, 1291–1305. [CrossRef] [PubMed]
65. Javed, M.A.; Younis, M.S.; Latif, S.; Qadir, J.; Baig, A. Community Detection in Networks: A Multidisciplinary Review. *J. Netw. Comput. Appl.* **2018**, *108*, 87–111. [CrossRef]