SLR-superscaffolder: a *de novo* scaffolding tool for synthetic long reads using a top-to-bottom scheme

Li Deng[a#], Lidong Guo[b,a], Mengyang Xu[a], Wenchao Wang[a], Shengqiang Gu[b], Xia Zhao[c], Fang Chen[c], Ou Wang[d], Xun Xu[d], Guangyi Fan[a#], and Xin Liu[a#]

[a]BGI-Qinqdao, BGI-SZ, Qingdao 266555, China

[b]BGI Education Center, University of Chinese Academy of Sciences, Shenzhen 518083, China

[c]MGI, BGI-Shenzhen, Shenzhen 518083, China

[d]BGI-Shenzhen, Shenzhen 518083, China

[#]Correspondence:

Xin Liu (liuxin@genomics.cn),

Guangyi Fan (fanguangyi@genomics.cn),

Li Deng(dengli1@genomics.cn)

Abstract

Synthetic long read (SLR) sequencing technologies, such as stLFR co-barcoded reads and 10X genomics linked-reads, have recently been developed and widely applied in genomics research. Here, we developed the SLR-superscaffolder, a standalone scaffolding tool for general synthetic long reads, with a top-to-bottom scheme where long fragment reads information is firstly used in large-scale scaffolding and followed by the paired-end information used in local scaffolding, to effectively use the SLR information. We tested SLR-superscaffolder power to assemble the human genome from three data resources. For instance, using the draft assembly with contig NG50 of 13 kb generated from 64-fold stLFR co-barcoded reads, SLR-superscaffolder significantly improved its scaffold NG50 to 15 Mb. Based on the draft assembly with scaffold NG50 of 58kb using 20-fold PCR-free NGS data, its scaffold NG50 was also drastically increased to 8 Mb. For the draft assembly with contig NG50 of 6.6 Mb from about 30-fold Oxford Nanopore long reads, SLR-superscaffolder presented a notable improvement in scaffold polishing with a scaffold NG50 of 21 Mb. Furthermore, comparing with other available SLR scaffolding tools, SLR-superscaffolder could produce an assembly with the highest quality of the longest contiguity and the least errors. Thanks to the valuable long-range information provided by SLR, SLR-superscaffolder shows a broad range of applications in the genome assembly. The source code is accessible on GitHub (https://github.com/BGI-Qingdao/SLR-superscaffolder).

I. Introduction

Single-tube long fragment read (stLFR) (Wang, et al., 2019), as one of the synthetic long read (SLR) technologies (Amini, et al., 2014; Kaper, et al., 2013; Peters, et al., 2012; Zheng, et al., 2016), has recently been developed for effectively barcoding long DNA fragments, pooling to construct single sequencing library, and sequencing using next generation sequencing (NGS) technologies. Since sequencing information for each long fragment is recoverable according to the same barcode shared by the reads from the specific long fragment, it can be applied to haplotyping (Amini, et al., 2014; Kuleshov, et al., 2014; Peters, et al., 2012; Zheng, et al., 2016), structural variation detection (Bishara, et al., 2015; Elyanow, et al., 2017; Marks, et al., 2019) and *de novo* genome assembly (Adey, et al., 2014; Coombe, et al., 2018; Kuleshov, et al., 2016; Weisenfeld, et al., 2017; Yeo, et al., 2017). Like the previous whole genome shotgun strategy of sequencing from a BAC (Gnerre, et al., 2011) and Fosmid library (Zhang, et al., 2012), stLFR can also retain the long-range information of high mass weight molecules, but it is more cost-effective to be widely applied in genome assembly, especially for some complex genomes.

As other SLR technologies (Amini, et al., 2014; Kaper, et al., 2013; Peters, et al., 2012; Zheng, et al., 2016) the coverage of barcoded reads for a single long fragment (LFR) in stLFR is too low to do a direct assembly by two stage processes (Bankevich and Pevzner, 2016) where the LFRs are separately assembled by reads with the same barcode, and then the genome is further assembled by the pre-assembled long LFRs. However, enhancement in LFR coverage of genome could overcome the weakness of current SLR technologies.

To utilize the LFR information, there have been several scaffolding tools previously developed for different SLR technologies, respectively. For contiguity preserving transposition sequencing data (CPT-seq), Adey, et al., developed FragScaff to do scaffolding using MST (minimum spanning tree) algorithm on the scaffold graph constructed by LFR information(Adey, et al., 2014). The scaffolder named Architect was designed for SLR sequencing technology from Illumina (Kaper, et al., 2013) by removing heuristically spurious edges on scaffold graph constructed by LFR information and paired-end (PE) information (Kuleshov, et al., 2016). For 10X Genomics Chromium technology (Zheng, et al., 2016), Warren, et al., developed a scaffolder with two versions ARCS (Yeo, et al., 2017) and ARKS (Coombe, et al., 2018) based on the long read scaffolder LINKS (Warren, et al., 2015), where the heuristic local extending algorithm was applied. In addition, Weisenfeld, et al., from 10X Genomics developed a *de novo* assembler named Supernova for Chromium sequencing data, which can assemble diploid genomes, instead of just scaffolding (Weisenfeld, et al., 2017). Recently, a universal assembler CloudSPAdes has been developed by Tolstoganov, et al., for SLR dataset to combine the assembly graph with LFR information based on SPAdes assembler (Bankevich, et al., 2012; Tolstoganov, et al., 2019). These currently available scaffolders do not explicitly consider the effects of misassemblies and non-unique properties of input contigs on scaffolding, while Supernova and CloudSPAdes do not provide an independent module for scaffolding. Thus, a universal scaffolder to deal with contigs assembled using various SLR data is required.

Here we proposed a standalone scaffolder (SLR-superscaffolder) by designing a top-to-bottom scheme with a screening algorithm for LFR information based on the statistical properties of SLR reads. As an independent scaffolder, the initial assembly (contigs or scaffolds) and SLR data are required which makes it convenient to be used with other tools and data. In the top-to-bottom scheme for using different information, the draft scaffolds are globally constructed using LFR information, and then the final delicate scaffolds are locally constructed using PE information. In the scheme of scaffolding, the order, orientation and gap size are determined in turn. In the screening algorithm implemented in the ordering process by LFR information, the minimum spanning tree (MST) and the heuristic pruning algorithms are combined to explicitly reduce the effect of input contigs with non-ideal seed contigs on scaffolding. In addition, the determinations of order and orientation of contigs are two individual modules in our scaffolder. Thus, the length requirement of input contigs is not as strict. All the above strategies make this scaffolder more robust to various inputs.

In development of the scaffolder, the statistical properties of SLR data are analyzed using stLFR data and reference of human genome. The results demonstrate that the stLFR co-barcoded reads technology refers to a general type of SLR dataset, where the number of LFR per barcode is significantly small due to the large number of barcodes in the library and the irregular distribution of insert size. We benchmark the SLR-superscaffolder and the other SLR scaffolders including FragScaff, Architect and ARKS using three different draft assemblies with stLFR co-barcoded reads of the human whole genome. The benchmark results show that the scaffolds assembled by SLR-superscaffolder are with longer contiguity and higher accuracy than others. Considering the generality of the stLFR co-barcoded reads in SLR, SLR-superscaffolder has great potential to be a universal scaffolder for various SLR datasets.

II. Method

As an independent scaffolding module, SLR-superscaffolder takes stLFR co-barcoded reads along with contigs/scaffolds assembled by any kind of datasets with any assembler as inputs. In our top-to-bottom scaffolding strategy, five independent modules are involved as shown in Figure 1, data preparing, ordering, orientating, local scaffolding, estimating of gap size.

Draft assembly resources

To assess the effectiveness of our algorithm, three draft assemblies of human whole genome (NA12878 cell line) have been used as input contigs/scaffolds, including contigs assembled by MaSuRCA (Zimin, et al., 2013) with stLFR co-barcoded reads only (MaSuRCA contigs), scaffolds assembled by SOAP*denovo2* (Luo, et al., 2012) with stLFR co-barcoded reads and additional 20-fold paired-end PCR-free NGS data (SOAP*denovo* scaffolds) and contigs assembled by Canu with about 30-fold ONT reads from (ONT contigs) Jain et al. work(Jain, et al., 2018). The evaluations for these input contigs/scaffolds have been listed in Table S1. The access information of these datasets is listed in Table S2. Except the ONT dataset, the new libraries for both stLFR and NGS were sequenced in this work and the information of all these datesets have been listed in Table S3. The stLFR library was constructed by MGIEasy stLFR Library Prep Kit and finally sequenced by BGISEQ-500. In constructing stLFR library, the long fragments longer than 50 kb were produced by shearing the subject's DNA, and then randomly trapped into barcoded magnetic beads in the single tube, and then fragmented into short sequences with the same barcode by two transposons (Wang, et al., 2019). The 20-fold pair-end NGS data with insert size about 390bp were generated without PCR amplification. The PCR-free library was constructed by MGIEasy FS PCR-Free DNA Library Prep Set V1.0 and sequenced by DNBSEQ-G400RS PE150.

Data preparing

To construct various scaffold graphs in following steps, the correlation between contigs/scaffolds is firstly derived from the alignments between contigs/scaffolds and stLFR co-barcoded reads. There are two different kinds of non-overlap information in stLFR co-barcoded reads, including PE information and LFR information (reads from one long fragment share the same barcode). In the SLR-superscaffolder, the BWA (Li and Durbin, 2009) is used to generate the alignment files, and both PE and LFR correlations are derived from the position of the aligned reads on contigs/scaffolds (Figure 1A). The data analysis of stLFR co-barcoded reads showed that the PE information is directed, with short correlation length, but the LFR information is undirected, with long correlation length. To effectively utilize the information, the long unique contigs are firstly scaffolded by the LFR information and then other contigs are locally scaffolded by the PE information. In practice, the contigs/scaffolds longer than a threshold with coverage around the average are considered as long unique contigs/scaffolds named seed contigs/scaffolds (Figure 1A).

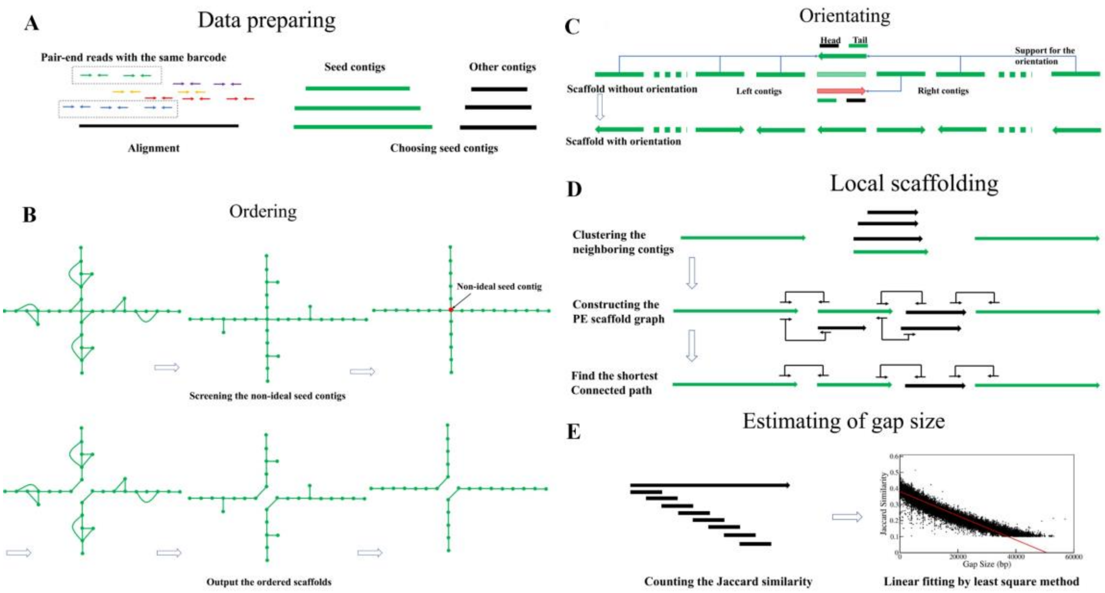
A
Data preparing
Pair-end reads with the same barcode
Seed contigs
Other contigs
Alignment
Choosing seed contigs
B
Ordering
Non-ideal seed contig
Screening the non-ideal seed contigs
Output the ordered scaffolds
C
Orientating
Head
Tail
Support for the orientation
Scaffold without orientation
Left contigs
Right contigs
Scaffold with orientation
D
Local scaffolding
Clustering the neighboring contigs
Constructing the PE scaffold graph
Find the shortest Connected path
E
Estimating of gap size
Jaccard Similarity
0.6
0.5
0.4
0.3
0.2
0.1
0
20000
40000
60000
Gap Size (bp)
Counting the Jaccard similarity
Linear fitting by least square method

Figure 1. SLR-superscaffolder algorithm. (A) In data preparing, two sub-processes are included: aligning the stLFR co-barcoded reads to the contigs/scaffolds and choosing the seed contigs/scaffolds. (B) In ordering, the suspicious seed contigs/scaffolds are screened interactively as shown in the upper three figures, and then the ordered scaffolds are generated as shown in the lower three figures. (C) In orientating, n-order neighboring contigs/scaffolds in ordered scaffold are used to determine the orientation of each contig/scaffold as shown in the upper figure, and the definitions of support for a given orientation of the contigs/scaffolds have been shown in the lower figure. (D) In local scaffolding, the contigs/scaffolds near a given neighboring contigs/scaffolds in the scaffold are clustered by LFR information firstly, and then the scaffold graph is further constructed by PE information. Finally, the shortest path between the neighboring contigs/scaffolds are output as the local scaffolds. (E) In the estimating of gap size, the statistical relation between similarity and distance is counted in the long contigs/scaffolds, and then an approximately linear relation is fitted using the least square method.

Ordering

A graph-based algorithm is applied to ordering the seed contigs (Figure 1B), requiring that an LFR scaffold graph should be constructed in advance. For all other SLR scaffolders, they assume that the closer two contigs are, the larger the number of shared barcodes would be. In this work, we use the Jaccard similarity (JS) of shared barcodes between two contigs to replace the number of shared barcodes. From the difference in the relations between JS and distance for stLFR co-barcoded reads and randomly barcoded reads in Figure S1, it is obvious that JS is able to determine the order and orientation among the contigs. Considering that the order and orientation are independently determined in SLR-superscaffolder, the vertices in LFR scaffold graph in ordering stage refer to seed contigs instead of the head/tail of contigs. The weight edge in the LFR scaffold graph is defined as the JS of shared barcodes between two contigs. To avoid the effect of contig length, the JS between contig *m* and contig *n* is defined as the maximum of JS between paired bins with the same length from each two contigs as following.

$$J(contig_m, contig_n) = \max\left(J\left(bin_i^m, bin_j^n\right)\right)\ for\ all\ pair(i,j)$$

where bin_i^m is the *i_{th}* bin in contig *m*. The Jaccard similarity between bins from different seed contigs is calculated as following.

$$J(bin_i^m, bin_j^n) = \frac{|barcodes(bin_i^m) \cap barcodes(bin_j^n)|}{|barcodes(bin_i^m) \cup barcodes(bin_j^n)|}$$

where $barcodes(bin_i^m)$ is the set of barcodes whose corresponding reads are aligned to the bin_i^m. In practice, the bins are chopped without any gaps or overlaps along the seed contigs, and a weight edge between two contigs is set as the JS only when it is larger than the threshold.

Similar to FragScaff (Adey, et al., 2014), the Prim's algorithm is applied to obtaining the maximum-weight minimum spanning tree (MST) of the LFR scaffold graph. Because the degree of branch of the maximum-weight MST is high, we are not able to efficiently order the seed contigs by extracting the trunk in the maximum-weight MST. The complexity of the maximum-weight MST comes from the non-ideal seed contigs, which lead to junctions with many long branches, in the seed contigs obtained in data preparing step. Meanwhile, the ideal seed contigs, long, unique and misassembly-free, would form a linear node or a junction with only two long branches in the maximum-weight MST. To decrease the complexity of the MST of the scaffold graph, we designed a simplifying strategy to distinguish the non-ideal seed from ideal seed contigs in the scaffold graph based on the different topologies in MST. To detect the non-ideal seed contigs, we firstly pruned the short tips in the MST, and then looked for all the junctions in the MST which are the non-ideal seed contigs. We updated the scaffold graph by screening vertices and edges related to the non-ideal seed contigs. Iteratively conducting the MST, detecting and screening steps for the scaffold graph, we efficiently decreased the fraction of non-ideal seed contigs in the scaffold graph. In practice, we set two control parameters to avoid the possible significant change in the connectivity of the scaffold graph by screening excessive seed contigs. Due to the reduction of the non-ideal seed contigs, the MST of the simplified scaffold graph has much lower degree of branch, and the long branches of the MST are output as the ordered scaffolds.

Orientating

Since the LFR information is undirected, it is impossible to determine the orientation between two contigs in the LFR scaffold graph using one JS. Similar to other scaffolders, the head/tail structure is utilized to determine the orientation of a contig in the ordered scaffolds by LFR information. But the head/tail structure is introduced after ordering in our strategy. The strategy has two advantages; one is reducing the effect of half-length in orienting stage on the ordering stage, the other is providing more local ordering information to the orienting step. Since the local order of contigs have been determined, the orientation of a contig in scaffold is not only determined by the correlation between two nearest neighboring contigs but also by the correlations between neighboring contigs with higher order as shown in Figure1C. According to the difference in the JS calculated with the head or tail of the specific contig, each neighboring contig can give a support for one orientation state as shown in Figure S2. After counting the number of supports for each orientation, the one with more supports will be chosen. This consensus strategy of orientating can make fully use of the local information and obtain more accurate orientation. Finally, the oriented scaffolds will be output.

Local scaffolding

In the above steps, the PE information of stLFR have not been used and only most of seed contigs have been ordered and oriented by LFR information. The unscaffolded contigs include the labeled non-seed contigs, short tips in maximum weight MST, and contigs screened in ordering. In this step, we insert the first two kinds of unscaffolded contigs into the oriented scaffolds by combining the PE and LFR information for each paired neighboring contigs in the oriented scaffolds one by one (Figure 1D). For one paired neighboring contigs, the unscaffolded contig, whose similarity to contigs in the gap is larger than a threshold, will be clustered as a candidate for local scaffolding. Using the PE information between these contigs, we construct a local directed scaffold graph, where the vertex set consists of candidate unscaffolded contigs and the paired neighboring contigs, and the directed edges refer to all the connections derived from PE information. Using depth-firstly search (DFS) algorithm on the directed graph, we search the shortest path between the paired neighboring contigs. If there is a connected path, the contigs on the path belong to the local scaffold according to the path information. The above process is similar to the scaffolding process of other assemblers, such as SOAP*denovo*, but the local scaffolding makes it more efficient to deal with the complex structure caused by the global non-unique contigs.

Estimating of gap size

Since the gap size information between neighboring contigs is useful for further analysis, we estimate the gap size using an empirical relation between their distance and similarity for the gaps formed with LFR information (Figure 1E). For the gap formed with PE information, the gap size is uniformly set to 11bp. Although the distance between two reads with the same barcode is unknown, a rough relation between the JS and the distance for two sequences is available according to the statistics from human stLFR data as shown in Figure S1. The relation is given by the statistics between sequences with a given size on the long contigs, with a linear fit using least square method. Finally, the gap sizes between two contigs are estimated by the similarity.

Evaluation

The standard metric of QUAST(Gurevich, et al., 2013) are used to evaluate the accuracy of the assembled results, where the effect of gap size is also considered. The accurate position of the sequence is determined by the alignment to the reference using Minimap2 (Li, 2018). According to the difference between relative positions of two consecutive sequences on the assembly and that on the reference, all the miassemblies are determined and a major misassembly is defined as an alignment of two consecutive sequences with difference larger than 1kb. The major miassemblies are further categorized as three types: relocation, inversion and translocation. Based on the different types of misassembly, we can roughly evaluate the accuracy of independent steps in our algorithm and compare with other scaffolding tools. The evaluation of QUAST was run with the parameters (*v5.0.2, -m 1000, -x 1000, -scaffolds, -scaffold-gap-max-size 10,000*).

III. Result and Discussion

The characters of stLFR co-barcoded reads

For SLR data, the number of long fragments per barcode and the insert size of PE reads are essential statistical properties in downstream analysis for the LFR and PE information. The insert sizes of PE reads were calculated by aligning paired reads to the reference genome. Similar to the strategy used to analyze the CPT-seq reads, and the distances between neighboring reads with the same barcode have been calculated after the aligned reads are sorted based on their genomic coordinates. Although the variation of gap size between two neighboring reads of one LFR is complex, the difference between the gaps between reads from the same long fragment and those between reads from two different long fragments can be distinguished by the statistical distribution of the gap size. There are three typical peaks, but the third peak is too low to be shown relative to the other peaks as seen in the inset (Figure 2a). The first peak corresponds to the gaps between paired reads from PE fragment, and the position of the peak indicates that the typical insert size of the PE fragment is about 251bp. The second peak corresponds to the gaps between neighboring reads from the same LFR, which is named intra-gaps, and the position of the peak indicates the typical size of intra-gap, about 2512bp. The third peak corresponds to the gaps between neighboring reads from two neighboring LFR with the same barcode, named inter-gaps, and the typical size of inter-gaps is about 50Mb. Compared with the gap size distribution of the CPT-seq reads (Adey, et al., 2014), the ratio of the peak value between inter-gaps and intra-gaps for stLFR co-barcoded reads is significantly lower, indicating that the average number of LFRs per barcode of stLFR co-barcoded reads is much smaller than that of CPT-seq reads. The number of physical partitions in stLFR library is about 50 million magnetic beads, which is much more than CPT-seq library and 10x Genomics Chromium library (Zheng, et al., 2016). The distribution of the insert size of stLFR co-barcoded reads is non-Gaussian (Figure 2b), which is different from that in the standard NGS library. All these results reveal that the properties of stLFR co-barcoded reads are more general than other kinds of SLR reads, and a more general scaffolding algorithm is required to efficiently exploit the PE and LFR information of the stLFR co-barcoded reads.

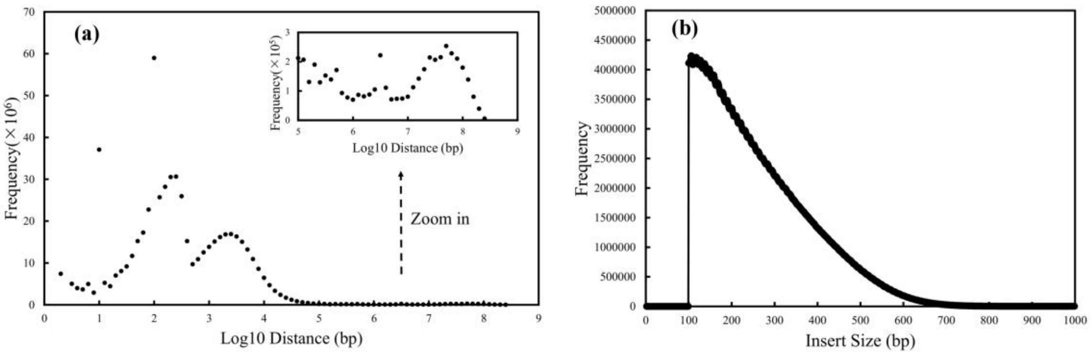

Figure 2. The distribution of gap size between neighboring reads with same barcodes (a) and the distribution of insert size of paired-end reads for stLFR co-barcoded reads (b).

Assembly results for different input contigs/scaffolds using different scaffolders

To evaluate the efficiency of SLR-superscaffolder using PE and LFR information of stLFR co-barcoded reads, we benchmarked SLR-superscaffolder and other SLR scaffolding tools including Fragscaff, Architect and ARKS using three input contigs/scaffold of human whole genome. All the results are listed in Table 1 and the run parameters for all the scaffolders are listed in Table S4. For other SLR scaffolders, the parameter sweeps have been completed for dataset of the human chromosome 19 (Chr19), and the optimal results have been listed in Table S5. Overall, the SLR-superscaffolder is more efficient to improve the contiguity of scaffolds using the LFR information in stLFR co-barcoded reads than other SLR scaffolders. The consumption of time in SLR-superscaffolder is also less than other SLR scaffolders, except ARKS which is alignment-free.

For the MaSuRCA contigs and SOAP*denovo* scaffolds, the contiguity of scaffolds assembled by the SLR-superscaffolder is the longest with the highest accuracy. Compared to MaSuRCA contigs in Table S1, the NG50 of scaffolds assembled by SLR-superscaffold is improved about 1317-fold from 13.4 kb to 17.6 Mb, and the NGA50 is improved about 29-fold from 13.2 kb to 380.5 kb. Compared to the SOAP*denovo* scaffolds in Table S1, the NG50 of scaffolds assembled by SLR-superscaffold is improved 227-fold from 40.1 kb to 9.1Mb, and the NGA50 is improved 44-fold from 34.3 kb to 1.5Mb. Except our tool, Fragscaff produces the scaffolds with the highest quality, where NG50 and NGA50 reach 400.9kb and 17.5 kb for the MaSuRCA contigs, and 2.3 Mb and 101.8 kb for the SOAP*denovo* scaffolds. In both cases, the contiguity and accuracy of scaffolds assembled by FragScaff, Architect and ARKS are significantly lower than those of the SLR-superscaffold which are also lower than those listed in the work of ARKS, since the original input contigs/scaffolds have shorter contiguity than those used in their previous work.

For the ONT contigs, the SLR-superscaffolder produces an improvement of contiguity about 3.3-fold from 6.6 Mb to 21.8 Mb， which is smaller than those by ARKS (about 6-fold) and FragScaff (about 4-fold), because the screening of contigs with misassemblies in our tool would decrease the number of links between long ONT contigs. As shown in the evaluation of the ONT contigs in Table S1, although the quality of the ONT contigs is high, the average number of misassemblies of a ONT contig is as large as about 3.2. Thus, for all the SLR scaffolds, the improvement of NGA50 is neglectable compared to that of the ONT contigs.

All the above results suggest that the LFR information of stLFR co-barcoded reads can be used to improve draft assembles using different assemblers with different datasets. Compared to the scaffolds assembled by MaSuRCA, where the LFR information has not been used, the scaffolds assembled by all the SLR scaffolders with stLFR co-barcoded reads are improved, especially for our SLR-superscaffolder. It is notable that SLR-superscaffolder can also make a greater improvement than other SLR scaffolders in terms of the input contigs with relatively high accuracy but short contiguity, indicating that SLR-superscaffolder is more robust to the quality of the inputs. Nevertheless, the accuracy of the input contigs/scaffolds is very important to the accuracy of the final scaffolds for all the SLR scaffolder. As a standalone scaffolder, the SLR-superscaffolder is more convenient to be used for exploiting the PE and LFR information of stLFR co-barcoded reads and combining with other different sequencing platform data.

Table 1. Evaluating summary of assemblies with different input contigs/scaffolds for human whole genome.

	stLFR scaffolds	FragScaff	Architect	ARKS	MaSuRCA
Human（masurca contig）					
Number of scaffolds (>1000bp)	119,630	166,476	307,205	213,076	300,831
Largest scaffold (bp)	62,433,675	4,043,217	202,889	2,756,390	177,746
Total assembled length (bp)	3,345,341,888	3,672,256,418	3,038,310,109	2,908,519,565	2,907,642,015
NG50 (bp)	17,657,864	400,954	14,042	37,406	13,405
NGA50 (bp)	380,495	17,539	13,705	15,020	13,232
Relocation	11,015	92,267	3,828	51,228	1,648
Inversion	2,939	5,349	1,637	5,190	180
Translocation	2,472	2,294	903	10,828	849
Number of misassemblies	16,426	99,910	6,368	67,246	4,475
Runtime	2day08h03min	3day20h47min	22day18h03min	18h25min	6day20h8min
Human（SOAP*denovo* scaffold）					
Number of scaffolds (>1000bp)	48,278	54,193	79,435	80,400	/
Largest scaffold (bp)	35,605,665	59,127,605	796,758	10,897,000	/
Total assembled length (bp)	3,115,923,941	3,094,665,921	2,659,717,897	2,713,878,926	/
NG50 (bp)	9,113,260	2,346,521	54,245	468,461	/
NGA50 (bp)	1,510,911	101,813	44,836	59,899	/
Relocation	2,373	32,588	1,104	20,906	/
Inversion	105	1,866	39	110	/
Translocation	3,694	2,926	875	4,060	/
Number of misassemblies	6,172	37,380	2,018	25,076	/
Runtime	1day18h08min	2day23h29min	10day16h53min	15h31min	/
Human（ONT contig）					
Number of scaffolds (>1000bp)	807	1,051	1,474	876	/
Largest scaffold (bp)	90,148,984	109,245,684	45,826,758	170,045,596	/
Total assembled length (bp)	2,829,830,390	2,828,106,943	2,823,722,824	2,823,836,148	/
NG50 (bp)	21,779,983	26,579,775	8,806,572	39,604,458	/
NGA50 (bp)	1,578,910	1,592,388	1,481,592	1,574,345	/
Relocation	4,378	4,182	3,962	4,266	/
Inversion	64	63	60	61	/
Translocation	1,575	1,453	1,383	1,607	/
Number of misassemblies	6,017	5,698	5,405	5,934	/
Runtime	4day08h38min	6day00h13min	2day22h42min	7h43min	/

Note: The runtime of MaSuRCA is for the whole assembly process not only the scaffolding process. SLR-superscaffolder, Fragscaff, ARKS and BWA were run using 8 threads, but Architect is single-thread.

Effects of length threshold of seed contigs

For the efficiency in the algorithm tests and parameter sweeps, the stLFR co-barcoded reads of Chr 19 have been extracted from the human whole genome dataset, and the input contigs are assembled by MaSuRCA only using the stLFR co-barcoded reads. Compared the scaffolding results of MaSuRCA with that of SLR-superscaffolder listed in Table S6, both the NG50 and NGA50 of scaffolds are substantially improved about 316-fold from 27.5 kb to 8.7 Mb and 33-fold from 26.3 kb to 873.7 kb, respectively. These results also demonstrate that both LFR and PE information is used in high efficiency and precision by SLR-superscaffolder.

The quality of input contigs has a strong effect on the scaffolding quality of the assembly, such as the contiguity and accuracy. The accuracy cannot be determined without a reference for the *de novo* assembly. Thus, we focus on evaluating the effects of contiguity by changing the length threshold to choose the seed contigs as shown in Figure 3. With the increase of length threshold, the scaffold NG50 monotonically decreases, while the NGA50 reaches a saturation peak between 5 kb and 10 kb. In terms of the major misassemblies, the numbers of inversion and relocation errors monotonically decrease with the increase of length threshold, and the decreasing rate in the region with smaller threshold is obviously higher than that in the region with larger threshold. The dependence of length threshold on the scaffold NG50 demonstrates that the connectivity of the LFR scaffold graph would be enhanced by involving more contigs when using a smaller threshold. However, the reduced misassembly number demonstrates that shorter contigs are easier to be misassembled. Thus, to get an optimal draft genome by LFR, it is very important to make a good balance between connectivity and proportion of short contigs. Although the balance does not only depend on the length threshold for the input contigs, the saturation peak of NGA50 indicates that our tool is robust to achieve a relatively optimal balance.

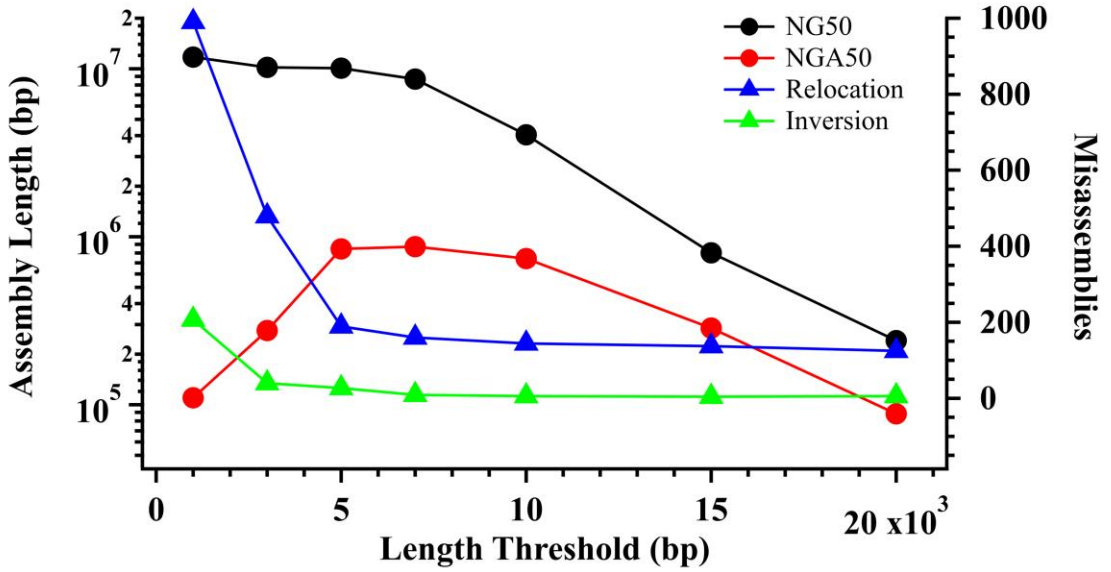

Figure 3. Quality of scaffolds assembled for different length thresholds of seed contigs.

In addition, the effects of local scaffolding by PE information and screening algorithm in ordering stage are tested using dataset of Chr19, listed in Table S6. Comparing the results with local scaffolding to those without, although the improvements of both contiguity and accuracy are not substantial, the decreasing ratio of inversion errors is as high as about 2.5-fold, indicating that the top-to-bottom scheme is an efficient way to take good use of complementary of PE information and LFR information. Comparing the results with screening to those without, both the scaffold NG50 and NGA50 of assembly with screening have substantial improvement, where the NG50 increases from 2.2 Mb to 8.7 Mb and the NGA50 from 661.9 kb to 873.7 kb, indicating that screening algorithm based on the graph theory in ordering can obviously improve the contiguity and accuracy of scaffold. By aligning the screened contigs to the reference, the properties of the screened contigs are analyzed in Table S7. According to the above results, contigs with misassemblies, high repeat content and short length, which increase the complexity of the LFR scaffold graph, are efficiently screened by our algorithm. The screening of these non-ideal seed contigs make substantial improvements in final assembled scaffolds.

IV. Conclusion

In summary, according to the statistical properties of stLFR data by analyzing the raw dataset of human genome with reference, we introduced a new top-to-bottom scaffolding algorithm in SLR-superscaffolder, where the LFR information and PE information can be used complementarily. For the human whole genome with input contigs/scaffolds assembled by short reads, the SLR-superscaffolder can produce about hundreds-fold increased NG50 of scaffolds with higher accuracy relative to the draft assemblies. These results demonstrate that the LFR information from stLFR library can be used to improve the quality for *de novo* assembly using our tool for draft genomes assembled by different strategies with different sequencing datasets.

The SLR-superscaffolder is the first scaffolder that considers the effect of misassemblies in the input contigs/scaffolds and provides systematical screening of non-idea contigs on the scaffold graph by combining with the MST algorithm and the topology of junctions in the MST. The results show that the screened non-idea contigs are usually those with shorter length, or higher repeat degree or misassembled. Compared with other SLR scaffolders, the SLR-superscaffolder produces assemblies with higher contiguity and accuracy for different input contigs/scaffolds, indicating that our tool is more efficient and robust to use LFR information of stLFR co-barcoded reads. It is important to note that all other SLR scaffolders are specifically designed for one SLR library dataset other than stLFR co-barcoded reads and the parameters of each SLR scaffolder used in different datasets may not be optimal although parameter sweeps have been conducted for Chr19.

As an independent scaffolder, SLR-superscaffolder can improve the quality of assembly results from other kinds of library (such as standard NGS or SMRT libraries) using LFR information in stLFR co-barcoded reads. Although the SLR-superscaffold is initially designed for stLFR co-barcoded reads, the LFR information in other kinds of SLR datasets can also been exploited with an appropriate format conversion due to the general properties of stLFR co-barcoded reads. Furthermore, since our approach is highly modularized, each stage in the SLR-superscaffolder will be separately improved when combined with other kinds of sequencing datasets such as SMRT and mate-pair in our future work.

Acknowledgement

We would thank Hongmei Zhu, Yinlong Xie and many other BGI-Shenzhen employees for fruitful discussions in the development of SLR-superscaffolder.

Reference

Adey, A., *et al.* (2014) In vitro, long-range sequence information for de novo genome assembly via transposase contiguity, *Genome research*, **24**, 2041-2049.

Amini, S., *et al.* (2014) Haplotype-resolved whole-genome sequencing by contiguity-preserving transposition and combinatorial indexing, *Nature genetics*, **46**, 1343.

Bankevich, A., *et al.* (2012) SPAdes: a new genome assembly algorithm and its applications to single-cell sequencing, *Journal of computational biology*, **19**, 455-477.

Bankevich, A. and Pevzner, P.A. (2016) TruSPAdes: barcode assembly of TruSeq synthetic long reads, *Nature methods*, **13**, 248.

Bishara, A., *et al.* (2015) Read clouds uncover variation in complex regions of the human genome, *Genome research*, **25**, 1570-1580.

Coombe, L., *et al.* (2018) ARKS: chromosome-scale scaffolding of human genome drafts with linked read kmers, *BMC bioinformatics*, **19**, 234.

Elyanow, R., Wu, H.-T. and Raphael, B.J. (2017) Identifying structural variants using linked-read sequencing data, *Bioinformatics*, **34**, 353-360.

Gnerre, S., *et al.* (2011) High-quality draft assemblies of mammalian genomes from massively parallel sequence data, *Proceedings of the National Academy of Sciences*, **108**, 1513-1518.

Gurevich, A., *et al.* (2013) QUAST: quality assessment tool for genome assemblies, *Bioinformatics*, **29**, 1072-1075.

Jain, M., *et al.* (2018) Nanopore sequencing and assembly of a human genome with ultra-long reads, *Nature biotechnology*, **36**, 338.

Kaper, F., *et al.* (2013) Whole-genome haplotyping by dilution, amplification, and sequencing, *Proceedings of the National Academy of Sciences*, **110**, 5552-5557.

Kuleshov, V., Snyder, M.P. and Batzoglou, S. (2016) Genome assembly from synthetic long read clouds, *Bioinformatics*, **32**, i216-i224.

Kuleshov, V., *et al.* (2014) Whole-genome haplotyping using long reads and statistical methods, *Nature biotechnology*, **32**, 261.

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

Li, H. and Durbin, R. (2009) Fast and accurate short read alignment with Burrows–Wheeler transform, *bioinformatics*, **25**, 1754-1760.

Luo, R., *et al.* (2012) SOAPdenovo2: an empirically improved memory-efficient short-read de novo assembler, *Gigascience*, **1**, 18.

Marks, P., *et al.* (2019) Resolving the full spectrum of human genome variation using Linked-Reads, *Genome research*.

Peters, B.A., *et al.* (2012) Accurate whole-genome sequencing and haplotyping from 10 to 20 human cells, *Nature*, **487**, 190.

Tolstoganov, I., *et al.* (2019) cloudSPAdes: assembly of synthetic long reads using de Bruijn graphs, *Bioinformatics*, **35**, i61-i70.

Wang, O., *et al.* (2019) 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 research*, **29**, 798-808.

Warren, R.L., *et al.* (2015) LINKS: Scalable, alignment-free scaffolding of draft genomes with long reads, *GigaScience*, **4**, 35.

Weisenfeld, N.I., *et al.* (2017) Direct determination of diploid genome sequences, *Genome research*, **27**, 757-767.

Yeo, S., *et al.* (2017) ARCS: scaffolding genome drafts with linked reads, *Bioinformatics*, **34**, 725-731.

Zhang, G., *et al.* (2012) The oyster genome reveals stress adaptation and complexity of shell formation, *Nature*, **490**, 49.

Zheng, G.X., *et al.* (2016) Haplotyping germline and cancer genomes with high-throughput linked-read sequencing, *Nature biotechnology*, **34**, 303.

Zimin, A.V., *et al.* (2013) The MaSuRCA genome assembler, *Bioinformatics*, **29**, 2669-2677.