ConnectedReads: machine-learning optimized long-range genome analysis workflow for next-generation sequencing

Chung-Tsai Su[1,*], Sid Weng[1], Yun-Lung Li[1], and Ming-Tai Chang[1]

1. ATGENOMIX INC.

*Corresponding author: chungtsai.su@atgenomix.com

Abstract

Current human genome sequencing assays in both clinical and research settings primarily utilize short-read sequencing and apply resequencing pipelines to detect genetic variants. However, structural variant (SV) discovery remains a considerable challenge due to an incomplete reference genome, mapping errors and high sequence divergence. To overcome this challenge, we propose an efficient and effective whole-read assembly workflow with unsupervised graph mining algorithms on an Apache Spark large-scale data processing platform called ConnectedReads. By fully utilizing short-read data information, ConnectedReads is able to generate haplotype-resolved contigs and then streamline downstream pipelines to provide higher-resolution SV discovery than that provided by other methods, especially in N-gap regions. Furthermore, we demonstrate a cost-effective approach by leveraging ConnectedReads to investigate all spectra of genetic changes in population-scale studies.

Background

Whole-genome sequencing (WGS) is increasingly used in biomedical research, clinical, and personalized medicine applications to identify disease- and drug-associated genetic variants in humans, all with the goal of advancing precision medicine [1]. At present, next-generation sequencing (NGS, also called short-read sequencing (SRS)) is a well-established technology used to generate whole-genome data due to its high throughput and low cost [2]. Resequencing, especially of human samples, is one of the popular applications of NGS. This process maps raw reads against a reference genome and determines all kinds of genomic variations, including single nucleotide polymorphisms (SNPs) and indels as well as genetic rearrangements and copy-number variants (CNVs) [3]. However, a fundamental flaw in the resequencing pipeline is that it ignores the correlation between sequence reads; thus, resequencing does not fully and properly utilize sequence data and may generate inconsistent alignments, which make variant calling, especially structural variant (SV) calling, more complicated [4, 5]. Since the human reference genome is incomplete and contains many low-complexity regions, assembling sequence reads without reference bias would be a proper way to overcome the above challenges. Nonetheless, assembly-based approaches for WGS data suffer from several computational challenges, such as high computing resource requirements and long turnaround times.

In this article, we propose an efficient whole-read assembly workflow with unsupervised graph mining algorithms on an Apache Spark large-scale big data processing platform called ConnectedReads. By leveraging the in-memory cluster computing framework of Apache Spark [6], ConnectedReads takes less than 20 hours to assemble 30-fold human WGS data and generates corresponding long haplotype-resolved contigs for downstream analysis, such as read mapping, variant calling or phasing. To evaluate the performance of ConnectedReads, we use 68 high-confidence insertions in the NA12878 sample detected by svclassify as SV benchmarks [7]. To demonstrate the ability of ConnectedReads, three samples from different populations are used. Through ConnectedReads, we are able to investigate unique non-reference insertions (UNIs) and non-repetitive, non-reference (NRNR) sequences from population datasets [8, 9]. Furthermore, ConnectedReads provides high resolution for SVs, especially on insertions. In conclusion, ConnectedReads optimizes NGS reads to generate long haplotype-resolved contigs, not only reducing mapping error but also streamlining SV detection.

Results

Data preparation

To assess the utility of ConnectedReads, three WGS datasets from three different ethnic groups were selected from publicly available databases, as listed in Table 1, including NA12878 of European ancestry, NA24694 of Asian ancestry, and NA19240 of African ancestry. In addition, the samples were sequenced by three different Illumina platforms, namely, the NovaSeq 6000 (NA12878), HiSeq 2500 (NA24694), and HiSeq X Ten (NA19240) platforms. Therefore, we believe that these datasets are representative of the majority of read-world data.

Table 1. Description of WGS datasets

Sample	Platform	Coverage	Description	Source
NA12878	NovaSeq 6000	30X	HG001 (Population: CEU)	*a
NA24694	HiSeq 2500	30X	HG006, Father of The Han Chinese GIAB Trio	*b
NA19240	HiSeq X Ten	35X	Yoruba (Nigeria) (Population: YRI)	*c

Data sources:

*a https://www.ebi.ac.uk/ena/data/view/ERR2438055

*b https://www.ncbi.nlm.nih.gov/sra/SRX1388455

*c https://www.ncbi.nlm.nih.gov/sra/SRX4637790

Evaluations

The results for the datasets in Table 1 obtained by applying the ConnectedReads workflow with default settings are listed in Table 2. ConnectedReads is clearly able to reduce the number of contigs and total base pairs by more than 96% and 87%, respectively. Since ConnectedReads generates haplotype-resolved contigs based on paired-end information, any two contigs would not be assembled together without sufficient support for paired-end information or overlaps. Although ConnectedReads aims to construct more accurate haplotype-aware contigs rather than longer ones, it is usually able to construct several contigs of more than 30 Kbps. In addition, there are 1,402,511, 1,224,389 and 2,082,886 contigs of >=1 Kbps for NA12878, NA24694 and NA19240, respectively. Furthermore, the length of contigs is strongly correlated with coverage and read length. The deeper the coverage is, the longer the contigs that can be generated are.

Table 2. Description of the contigs of three datasets generated by ConnectedReads.

Sample	NA12878	NA24694	NA19240
Number of contigs	16,348,524	18,900,370	18,261,647
Total base pairs (bps)	10,466,069,046	9,925,886,082	14,093,570,587
Average length	640	525	772
Longest contig	37,619	33,904	32,145
# of singletons (<=151)	8,351,671	11,305,117	8,355,394
>=1 Kbps	2,899,483	2,470,630	4,487,568
>=2 Kbps	1,402,551	1,224,389	2,082,886
>=5 Kbps	263,878	256,178	277,357

ConnectedReads offers two advantages for downstream analysis. One is mapping recovery, and the other is SV detectability. Both of these advantages are described comprehensively by the following experimental results and case studies.

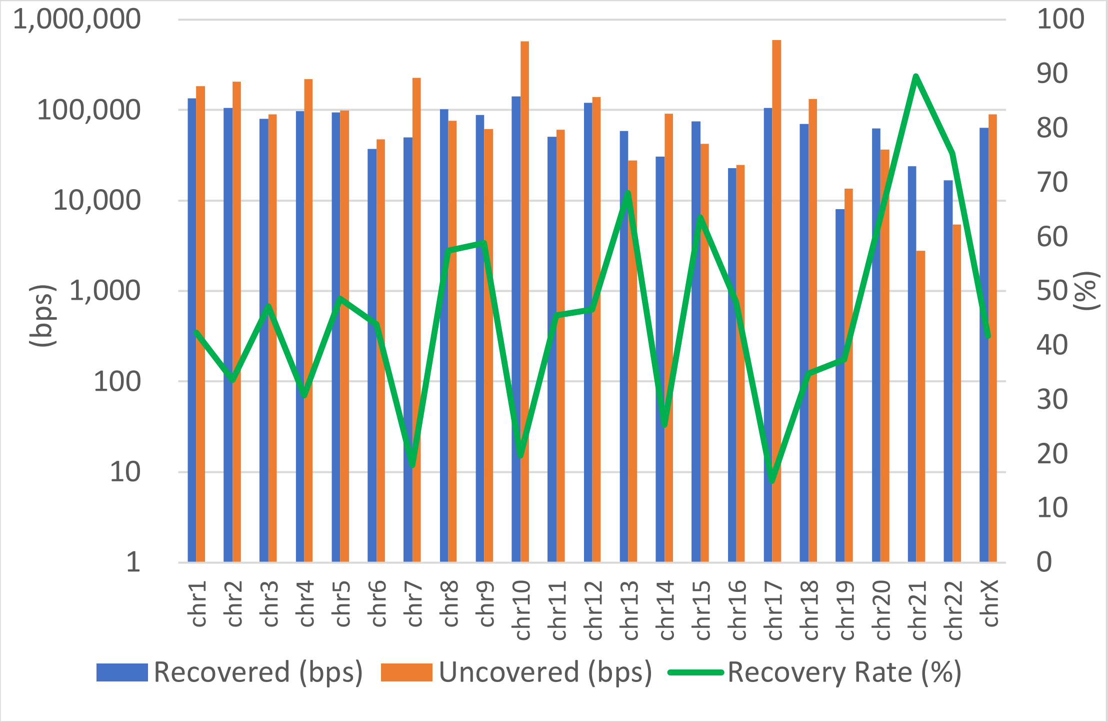


Fig. 1. Mapping recovery of NA12878 (short reads vs. ConnectedReads). BWA and minimap2 are adopted for the short-read data and ConnectedReads contigs, respectively. Recovery Rate = # recovered / (# recovered + # uncovered).

First, mapping recovery is the best way to evaluate the advantage of ConnectedReads. According to the evolution of NGS technology in recent years, longer reads can reduce two kinds of mapping errors, namely, false mapping and uncovered regions based on the reference genome [10]. However, it is hard to determine whether a mapping record is false because of several complicated situations, such as sequencing errors, an incomplete reference genome, high sequence divergence and SVs [5]. Therefore, the recovery rate is a proper measurement for evaluating the performance of ConnectedReads and SRS. Recovery refers to the regions of the reference genome that have no short reads mapped by using Genome Analysis Toolkit (GATK) Best Practices (i.e., BWA-MEM) [11] but have mapping records in the ConnectedReads dataset by using minimap2 [12]. In terms of NA12878 mapping recovery, as illustrated in Fig. 1, there is a 15%-90% recovery rate for each chromosome (excluding chrY). For example, chr1 has 319,018 uncovered bps with SRS, but 135,253 bps (42.4%) can be recovered by ConnectedReads. The best recovery rate (89.6%) is on chr21, and the worst recovery rate (15.4%) is on chr17. Large regions are often recovered when large deletions occur. Overall, ConnectedReads is able to reconstruct the mapping information for uncovered regions in the reference genome by using SRS data, and it may be the best candidate complementary to Illumina short-read data.

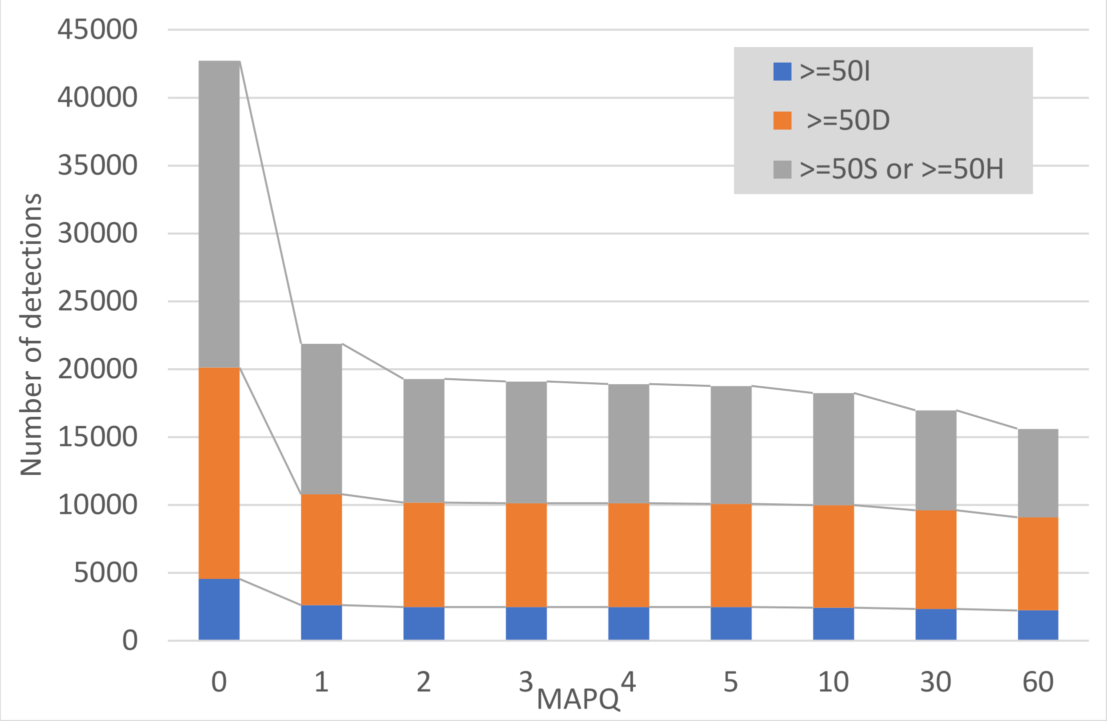


Fig. 2. Number of SV vs. MAPQ (NA12878). The priority order used for counting is insertion > deletion > soft-/hard-clipped sequence. Any two SVs will be merged if their distance is less than 50 bps. This means that if one insertion and one deletion are encountered in the same location, only one insertion is counted.

Second, SV detection remains a challenge in SRS [13]. Using ConnectedReads technology will significantly mitigate this challenge because ConnectedReads has the same competitive advantage as long-read sequencing (LRS) from Nanopore [14] and PacBio [15]. The longer the read is, the more correct the mapping result is and the more easily SVs can be found. Fig. 2 shows the numbers of insertions and deletions identified in NA12878. The method is simply based on a CIGAR string generated by minimap2. When the threshold of mapping quality (MAPQ) is increased, fewer insertions and detections are identified. Since several alignment records with an MAPQ of 0 were falsely mapped in our investigation, the threshold of MAPQ was set to 1 in the following experiments to balance the precision and sensitivity of SV detection. In addition, several recent studies have revealed that every human genome has approximately 20,000 SVs that span at least 10 million bps [16, 17]. ConnectedReads identifies 21,855 SVs, and the number of SVs is similar to that obtained in previous studies [16, 17].

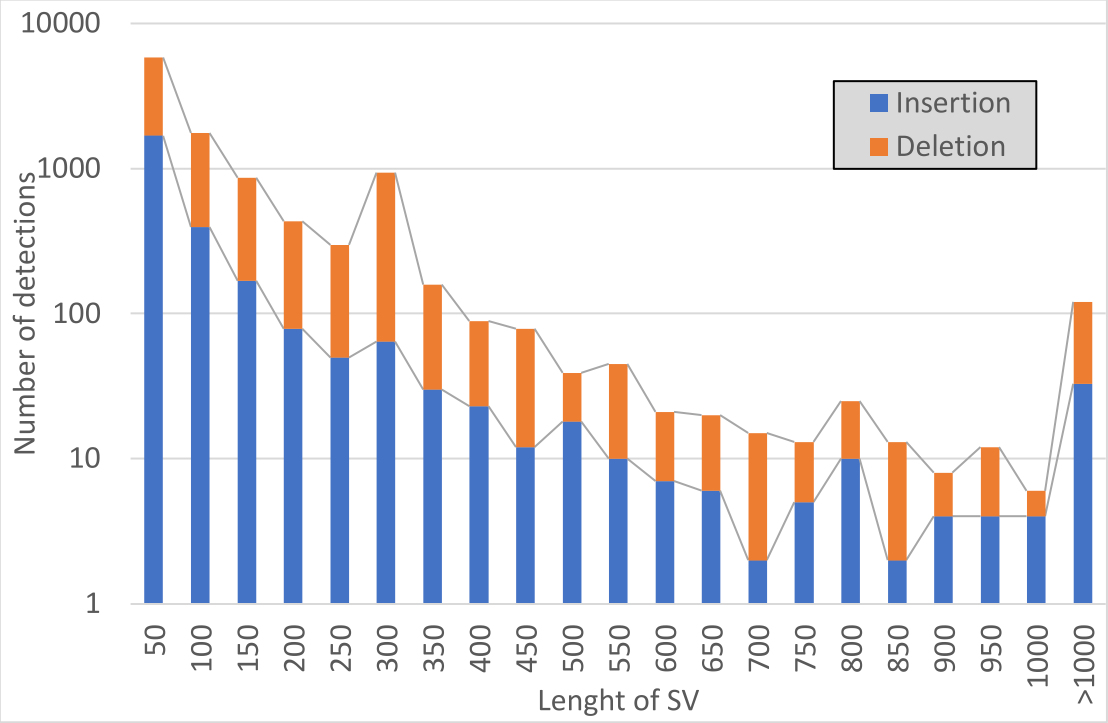


Fig. 3. SV length distribution of NA12878. The peak at a length of 251-300 is attributed primarily from Alu elements.

Furthermore, an interesting phenomenon is observed when the insertions and deletions shown in Fig. 2 are sorted by length (Fig. 3). Based on the 1000 Genomes Project and several studies [18, 19], the number of SVs decreases as the length of the SVs increases. Therefore, the majority of the SVs are small indels (<50 bps). Then, the trend of the distribution slightly decreases as the length of the SVs increases, except for the peak at 250-300 bps. This change is due to abundant Alu elements whose body lengths are approximately 280 bases [20]. In addition, several studies have reported this phenomenon when using PacBio LRS [16]. Therefore, ConnectedReads is able to complement SRS in not only mapping coverage but also SV detection.

Comparison against a high-confidence truth set

To evaluate the sensitivity of ConnectedReads in SV detection, the benchmark dataset collected by svclassify [7] is used. Since insertions are more difficult to detect than deletions, the 68 high-confidence insertions from svclassify are chosen as the insertion benchmarks in this paper. Additionally, two well-known variant callers are selected for performance evaluation: pbsv [21] for PacBio LRS and FermiKit [22] for Illumina SRS. Both FermiKit and ConnectedReads adopt assembly-based approaches to prevent reference bias. The results for the insertion benchmark data are listed in Table 3. PacBio LRS data are usually able to cover the whole region of most SVs, so pbsv achieves a 91.2% detection rate. However, FermiKit detects only 28 insertions since it aims to construct the complete sequence for each insertion. Since most WGS samples are sequenced by SRS and have a coverage of approximately 30X, it is quite hard to reconstruct the complete sequence (including novel insertions) through de novo assembly. Therefore, ConnectedReads provides a naïve insertion caller to relax the constraint by proposing three levels of detection, namely, completely detected, partially detected, and potentially detected. The ConnectedReads^{INS} caller has a strict criterion because it is based on both completely detected and partially detected insertions. The ConnectedReads^{SV} caller has a lenient criterion because it accommodates all three levels of detection. As shown in Table 3, ConnectedReads^{INS} and ConnectedReads^{SV} achieve 86.8% and 95.6% detection rates, respectively, indicating that ConnectedReads is able to achieve the same level of insertion-detection performance as PacBio LRS. Therefore, the above experimental results give us confidence to investigate SVs in population-scale data.

Table 3. Long insertion benchmarks in NA12878 from svclassify

Method	Detection rate (%)	Number of insertions with complete sequences
pbsv (PacBio)*	91.2	62
FermiKit	41.2	28
ConnectedReads^{INS}	86.8	32
ConnectedReads^{SV}	95.6	32

*The result of pbsv from NIST's Genome in a Bottle (GIAB) project is available at ftp://ftp-trace.ncbi.nlm.nih.gov/giab/ftp/data/NA12878/analysis/PacBio_pbsv_05212019/HG001_GRCh38.pbsv.vcf.gz

Therefore, ConnectedReads is competent at SV detection, especially for insertions. Then, the next question that we are interested in is how many unique SVs exist in each population. According to Kehr's findings (in Table S4 of [9]), there are 372 SVs in all Icelanders. After removing redundant SVs and merging adjacent SVs, 248 distinct SVs are represented as the second set of benchmark data in this paper. We are eager to know whether these 248 SVs are unique to Icelanders or shared by all populations. Therefore, the three samples from different continents shown in Table 1 are processed by ConnectedReads, and then minimap2 is used to generate alignment records for their ConnectedReads contigs. Then, we manually evaluate whether the 248 SVs exist in the three samples and show the result in Table 4. More than 95% of the SVs found in all Icelanders are found in the three different populations, and only one SV cannot be found in any of the three given samples. It is obvious that most common SVs in Icelanders are not unique to Icelanders. Additionally, approximately 40 SVs should not be classified as SVs in the given samples because they are composed of multiple small variants. More details will be discussed in the next section. It is believed that ConnectedReads provides us with better resolution than other tools to observe SVs.

Table 4. The SV benchmarks from all Icelanders.

Sample	Race	Detected		Undetected	
		SV	Multiple small variants	Not found	Uncovered
NA12878	Caucasian	204	39	2	3
NA24694	Asian	198	40	2	8
NA19240	African	205	39	1	3

Discussion

Accuracy of insertion length

In the above section, ConnectedReads not only was complementary to the resequencing pipeline of SRS in terms of uncovered regions but also was able to detect SVs, especially long insertions. For example, at least 85% of long insertions can be detected by ConnectedReads, and 32 insertions are completely constructed, as shown in Table 3. Interestingly, most of the insertion sequences constructed by ConnectedReads are shorter than those in the report provided by svclassify. Since svclassify leverages Spiral Genetics to identify the 68 high-confidence insertions, PacBio LRS is adopted as an independent reference. As shown in Fig. 4(a), most of the insertions identified by ConnectedReads are apparently shorter than those identified by Spiral Genetics. However, Fig. 4(b) shows that 24 of 32 insertions have identical lengths when identified by PacBio and ConnectedReads. In addition, the remaining insertions have only slight differences. The difference between the results from svclassify and ConnectedReads could have two major causes. First, sequencing-related issues, including wet-laboratory processes and sequencing platforms, should be the main cause. Second, the different data analysis pipelines will lead to different results, especially when using different genome references. ConnectedReads and PacBio adopt HG38, but svclassify uses HG19. For comparison, the insertions provided by Spiral Genetics are transferred to HG38 by using UCSC LiftOver [23]. The step for transforming the coordinates might also lead to inconsistencies. Regardless, ConnectedReads with SRS can construct complete and accurate insertion sequences as well as PacBio LRS can.

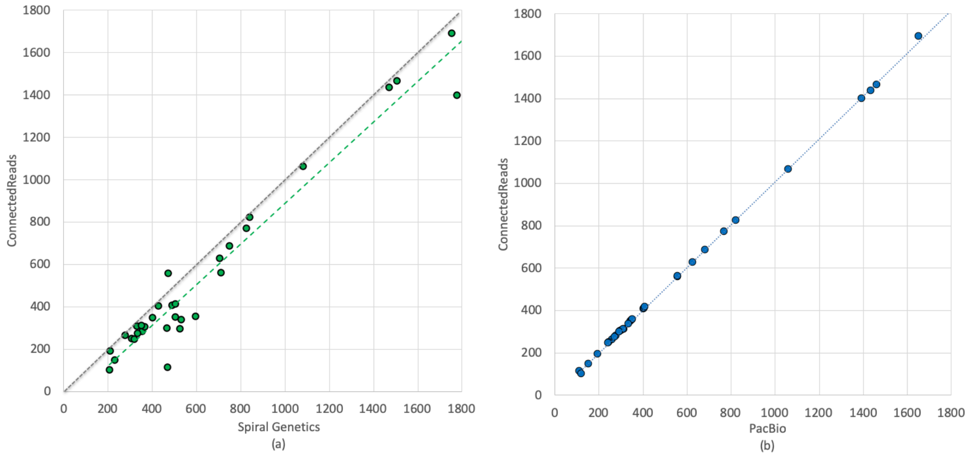
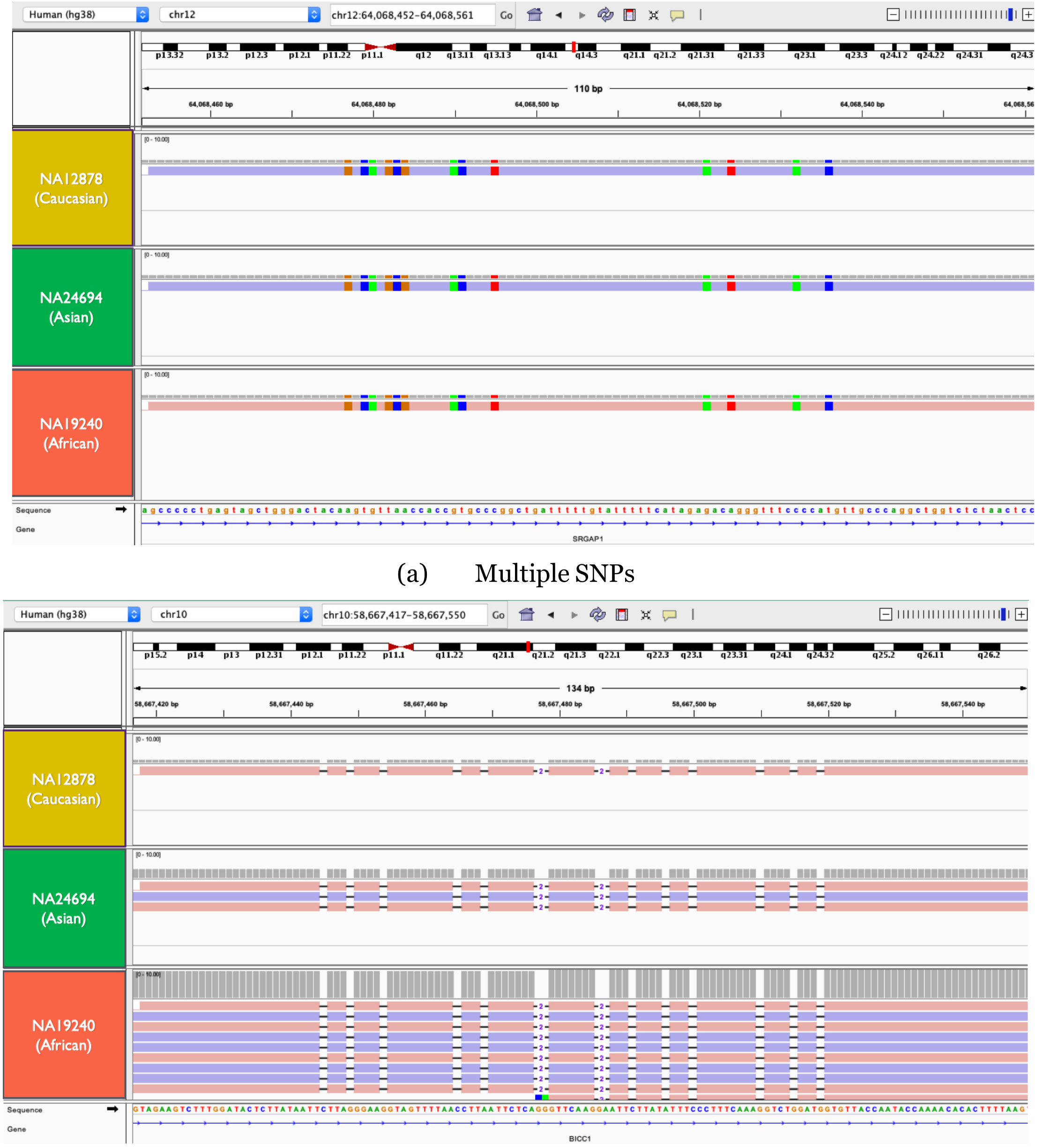


Fig. 4. Comparison of insertion lengths among three approaches. (a) Spiral Genetics vs. ConnectedReads, (b) PacBio vs. ConnectedReads. The gray dashed line is the 1:1 line. The green and blue dashed lines in (a) and (b), respectively, represent the moving average of the comparison.

Granularity of SV detection

Another interesting phenomenon shown in Table 4 caught our attention. Although common SVs in Icelanders exist in different populations, approximately 40 of the SVs should not be classified as SVs (≥ 50 bps) because they are composed of multiple small variants, as illustrated in Fig. 5. Fig. 5(a) and Fig. 5(b) contain only 13 SNPs and 13 deletions spanning 60 bps and 80 bps, respectively. When many adjacent variants occur in any individual, most mapping tools have limited information with which to correctly arrange reads with many adjacent variants and then straightforwardly choose to either employ soft/hard clipping or categorize them as unmapped. This limitation will somehow guide most variant callers to identify these regions as SVs. ConnectedReads can prevent such false mapping and help variant callers detect the adjacent variants correctly and precisely.

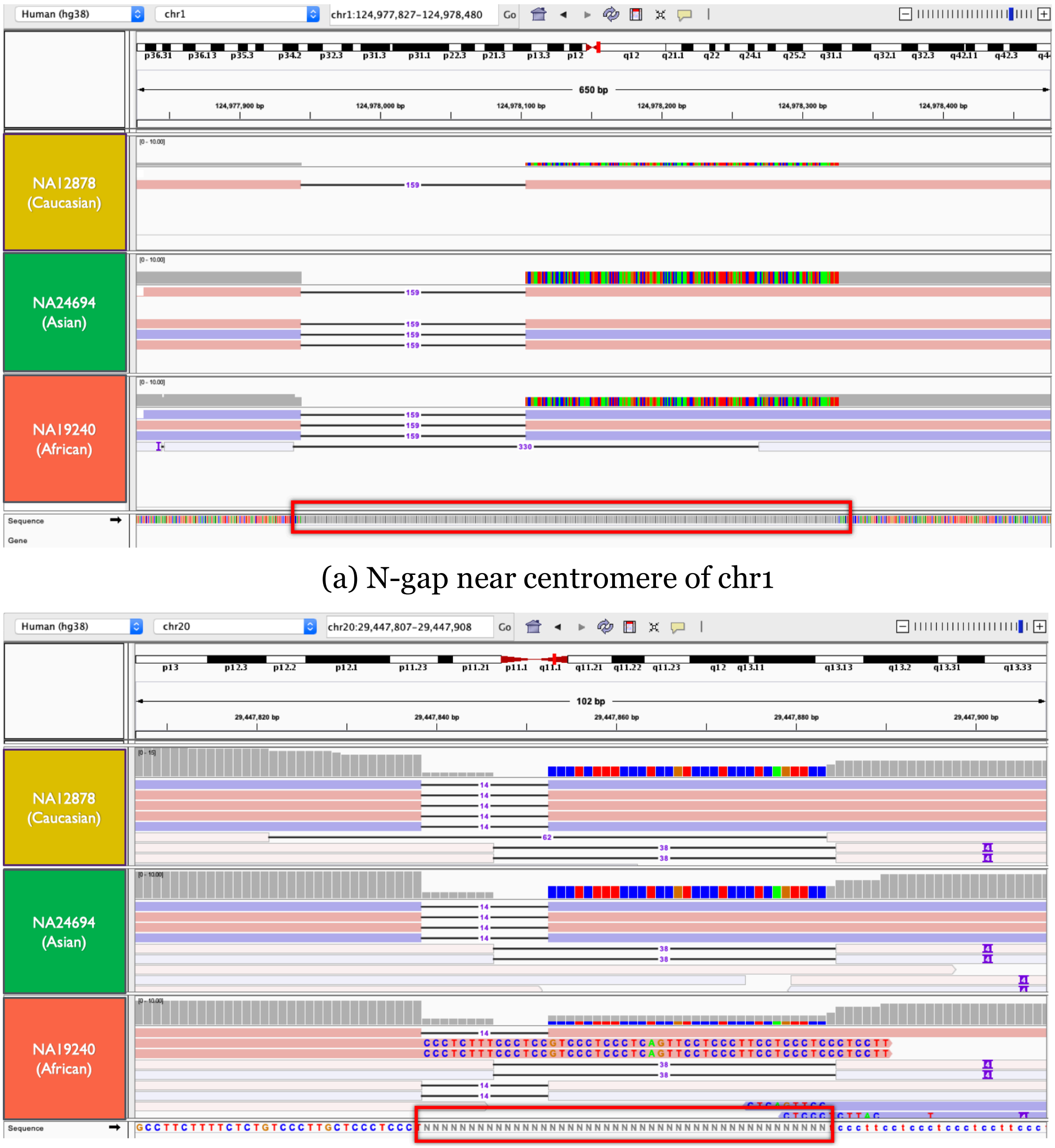


(a) Multiple SNPs

(b) Multiple deletions

Fig. 5. Examples of NRNR sequences (multiple small variants). (a) Thirteen adjacent SNPs in the intron of SRGAP1. (b) Thirteen adjacent deletions in the intron of BICC1.

Furthermore, ConnectedReads is able to mitigate the impact on downstream analysis due to the incomplete human reference genome. The most recent human genome still has many ambiguous areas (N-gaps), and they are mainly located in centromeres and telomeres. Fig. 6 illustrates that two ambiguous gaps can be assembled by using ConnectedReads and that the sequences are totally identical among the three individuals from different populations. This finding gives us strong confidence that most humans might have the same sequence in the two N-gap regions. By randomly selecting two Chinese adults, the occurrence of identical sequences in the N-gap regions is confirmed by Sanger sequencing, as performed by a Clinical Laboratory Improvement Amendments (CLIA)-certificated laboratory. The length of the ambiguous region in Fig. 6(a) should be corrected from 382 to 223 bps. In addition, the length of ambiguous regions in Fig. 6(b) should be shortened from 45 to 31. Based on these cases, ConnectedReads is able to provide a cost-effective approach with which to complete the human reference genome.



(a) N-gap near centromere of chr1

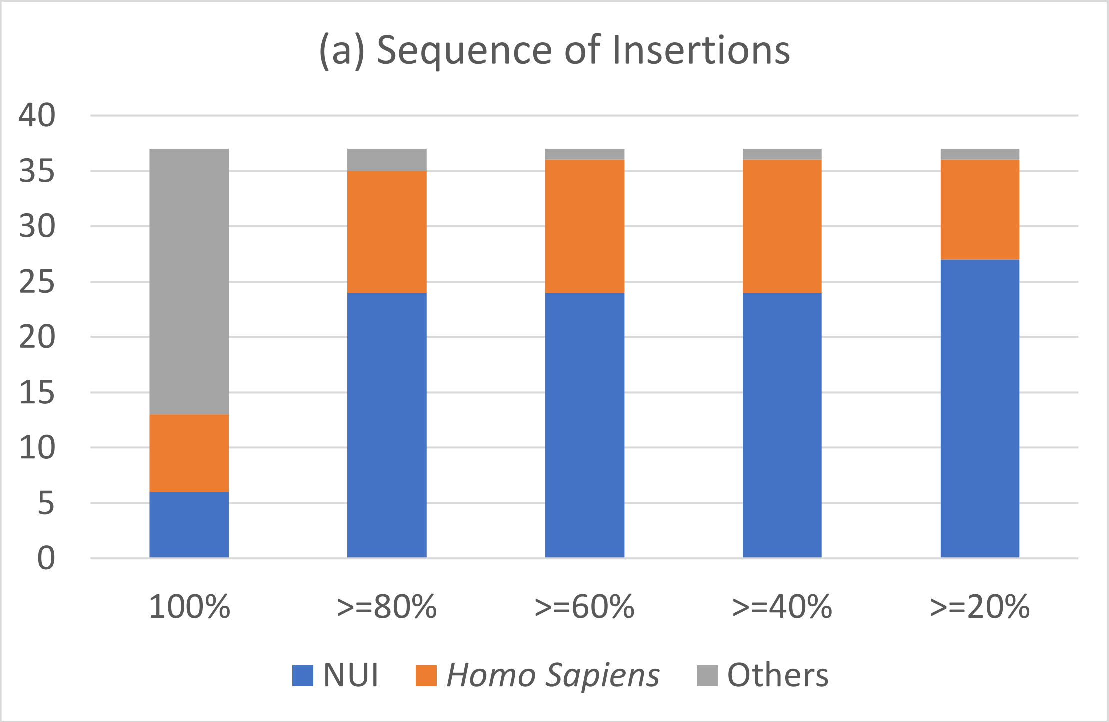
(b) N-gap near centromere of chr20

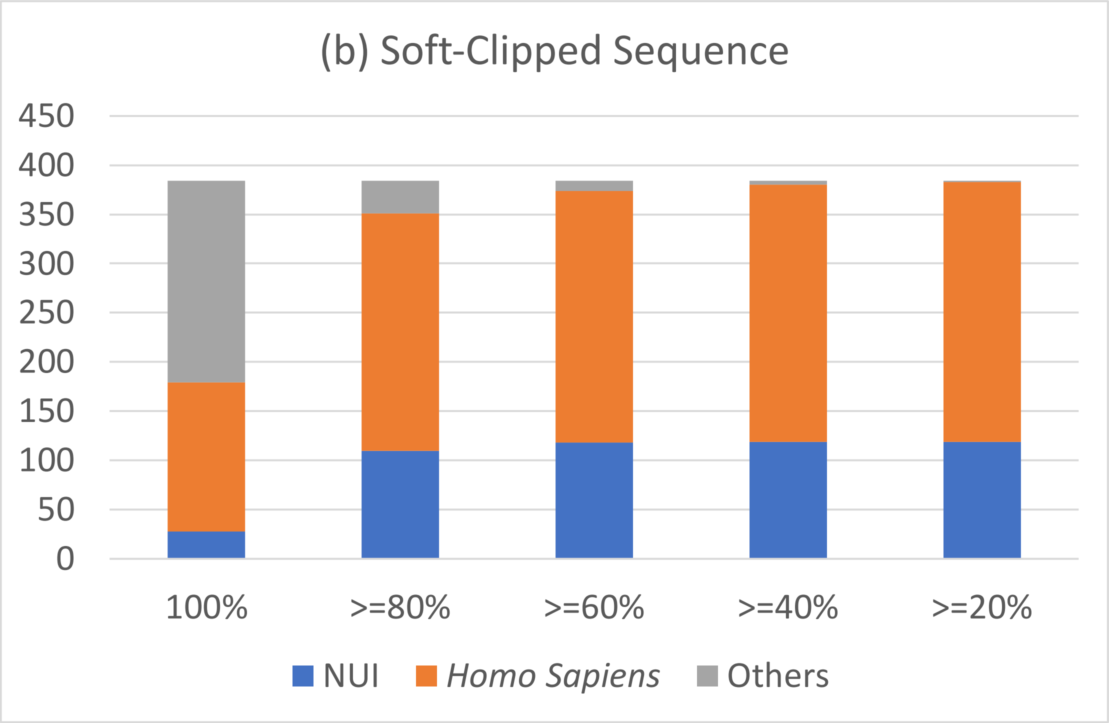
Fig. 6. The contigs from three individuals cross the entire N-gap regions near the centromere of (a) chromosome 1 and (b) chromosome 20. Based on the alignment result, the sequences in the N-gap are identical among

the three individuals. (a) The genomic location of IGV is chr1:124,977,874-124,978,413 of HG38. (b) The genomic location of IGV is chr20:29,447,807-29,447,908 of HG38.

Correctness of sequence assembly

The next topic for the evaluation of ConnectedReads is correctness. To comprehensively investigate the sequence assembly correctness of ConnectedReads, insertion sequences, soft-clipped sequences and unmapped contigs are selected and then identified as being from *Homo sapiens* or just chimeric DNA sequences resulting from false reconstruction. After removing sequences with a length < 1,000 bps, there are 37, 384 and 411 insertions, soft-clipped sequences and unmapped contigs, respectively. By using BLAST to find any homologous sequences in the National Center for Biotechnology Information (NCBI) non-redundant sequence (nr) database, each sequence can be identified as human or not. As shown in Fig. 7, 35.1%, 46.6% and 37.7% of the insertions, soft-clipped sequences and unmapped contigs are identical to *Homo sapiens* DNA sequences in the nr database. As the threshold of similarity is continuously lowered, more evidence can be found to support the sequence assembly correctness of ConnectedReads. However, there are six sequences without any support when the threshold is set to 20%. Two of the contigs have low-complexity content, and two are matched to several entries but with less than 10% support. The last two unmapped contigs (CONTIG-8337086 and CONTIG-15793805) are more than 10 Kbps in length. CONTIG-15793805 covers all of CONTIG-8337086 in reverse-complement mode, so CONTIG-15793805 is represented and shown in Fig. 8. CONTIG-15793805 is almost fully covered by several *Homo sapiens* sequences, some of which overlap. Therefore, these non-reference contigs constructed by ConnectedReads are all from *Homo sapiens*. Based on the above findings, ConnectedReads achieves high data correctness.


(a) Sequence of Insertions
40
35
30
25
20
15
10
5
0
100%
>=80%
>=60%
>=40%
>=20%
NUI
Homo Sapiens
Others


(b) Soft-Clipped Sequence
450
400
350
300
250
200
150
100
50
0
100%
>=80%
>=60%
>=40%
>=20%
NUI
Homo Sapiens
Others

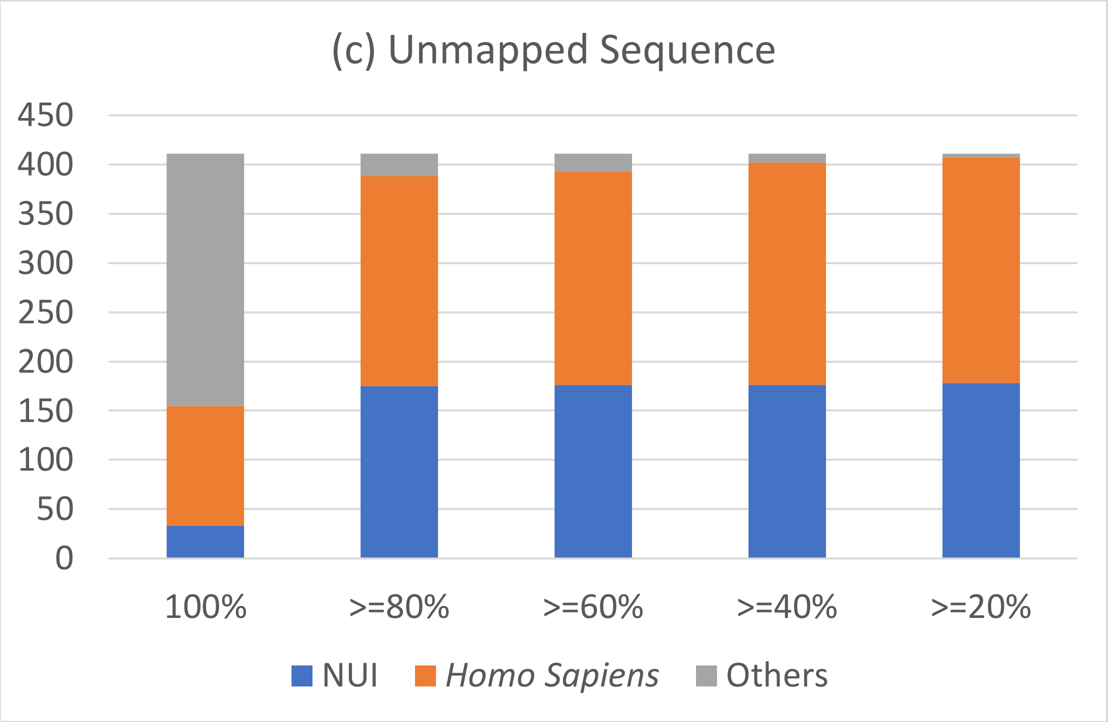
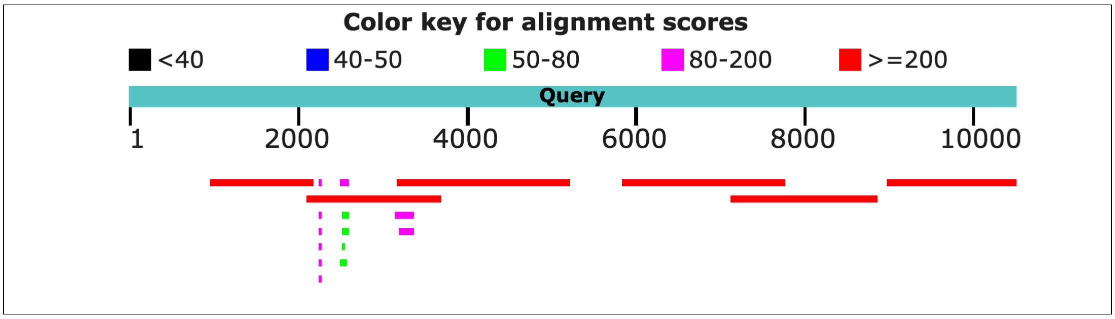
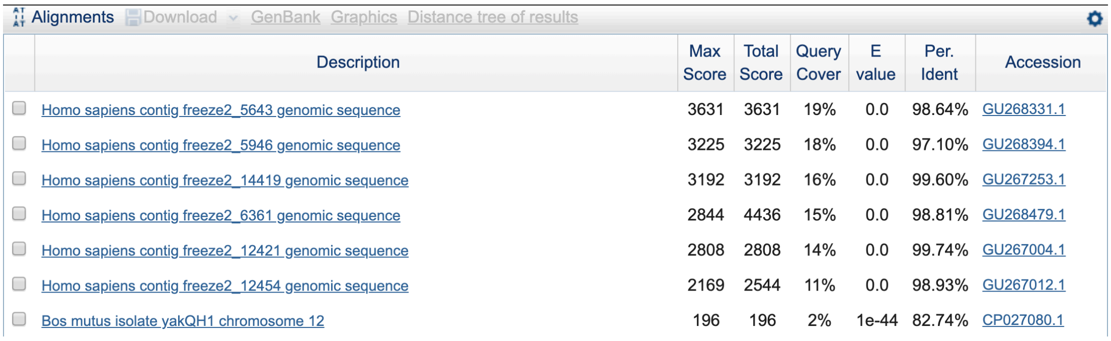


Fig. 7. Distribution of BLAST queries for non-reference sequences identified by ConnectedReads. (a) From insertions, (b) from soft-clipped sequences and (c) from unmapped contigs. The x-axis shows the similarity of the query results by BLAST. NUI means that the matched result is annotated as a non-reference unique insertion. Homo sapiens means that the matched result is from Homo sapiens or Human BAC.





	Description	Max Score	Total Score	Query Cover	E value	Per. Ident	Accession
□	Homo sapiens contig freeze2_5643 genomic sequence	3631	3631	19%	0.0	98.64%	GU268331.1
□	Homo sapiens contig freeze2_5946 genomic sequence	3225	3225	18%	0.0	97.10%	GU268394.1
□	Homo sapiens contig freeze2_14419 genomic sequence	3192	3192	16%	0.0	99.60%	GU267253.1
□	Homo sapiens contig freeze2_6361 genomic sequence	2844	4436	15%	0.0	98.81%	GU268479.1
□	Homo sapiens contig freeze2_12421 genomic sequence	2808	2808	14%	0.0	99.74%	GU267004.1
□	Homo sapiens contig freeze2_12454 genomic sequence	2169	2544	11%	0.0	98.93%	GU267012.1
□	Bos mutus isolate yakQH1 chromosome 12	196	196	2%	1e-44	82.74%	CP027080.1

Fig. 8. BLAST result for CONTIG-15793805. Six long *Homo sapiens* contigs are identified with low E-values. Since some of them overlap, the alignment result gives us strong confidence in the data correctness of ConnectedReads.

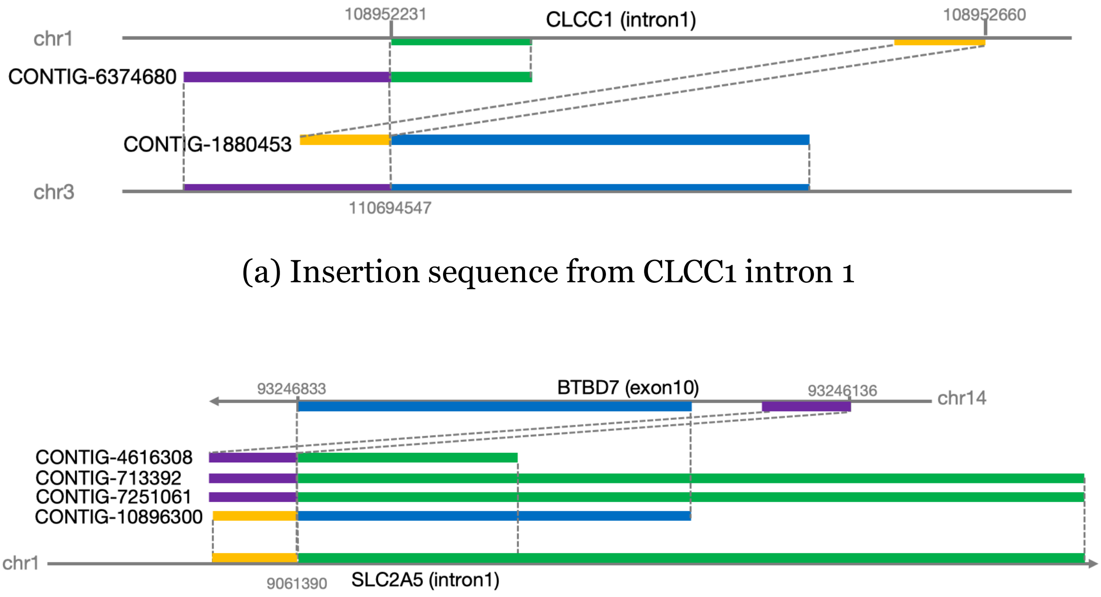
Translocation-based insertions

Furthermore, another way to evaluate the data correctness of ConnectedReads is to check whether there is any translocation. If any two unrelated sequences are incorrectly assembled together, it will cause a fake translocation event, in which a contig is mapped to multiple chromosomes. Table 5 lists the 13,686 contigs with multiple alignment records on two chromosomes in sample NA12878. After removing singletons and filtering out low-MAPQ records (MAPQ < 60), 71 qualitied contigs are represented by 22 translocation groups.

Table 5. Translocation-based insertions in NA12878.

Item	Number
Number of contigs mapped on at least two chromosomes	13,686
Number of contigs after removing singletons	2,075
Number of qualified contigs	71
Number of high-confidence translocation groups	22
Number of translocation-based insertions	8

Interestingly, eight translocation groups have one clear breakpoint on one chromosome but two breakpoints on another chromosome. As shown in Fig. 9(a), for example, CONTIG-6374680 and CONTIG-1880453 have identical breakpoints at chr3:110694547. However, CONTIG-6374680 and CONTIG-1880453 have their own breakpoints at chr1:108952231 and chr1:108952660, respectively. Thus, in NA12878, the 430-bp sequence in intron 1 of CLCC1 is inserted at chr3:110694547. Fig. 9(b) also shows that the 697-bp sequence in exon 10 of BTBD7 is inserted into intron 1 of SLC2A5. These translocation groups are called translocation-based insertions. ConnectedReads proposes a naïve way to investigate these translocation-based insertions. In summary, ConnectedReads not only constructs genome sequences precisely but also facilitates SV detection by mitigating reference mapping bias.



(a) Insertion sequence from CLCC1 intron 1

(b) Insertion sequence from BTBD7 exon 10

Fig. 9. Examples of translocation-based insertions. (a) The 429-bp sequence from intron 1 of CLCC1 is inserted at chr3:110,694,547. (b) The 697-bp sequence from exon 10 of BTBD7 is inserted into intron 1 of SLC2A5.

Conclusions

ConnectedReads leverages SRS to generate long haplotype-resolved contigs such as those produced by 3rd-generation sequencing technologies (i.e., PacBio and Nanopore) to not only prevent mapping errors but also facilitate SV discovery. In summary, ConnectedReads can serve as an NGS gateway for streamlining downstream data analysis, such as false positive prevention, SV detection, and haplotype identification.

Materials and methods

Workflow

ConnectedReads leverages Apache Spark [6], a distributed in-memory computing framework, to accelerate its whole workflow, as illustrated in Fig. 10. To fully utilize the power of the distributed framework, the preprocessing step involves splitting a large compressed file into several small files and then uploading those files into the Hadoop distributed file system (HDFS) since most of the WGS samples exist in two separate FASTQ files in GZIP format. First, we adopt Apache Adam [24], as shown in Fig. 10(a), to transform data from FASTQ format to column-based Parquet format for data parallel access. To facilitate data processing in the following steps, we extend Adam to not only encode paired-end information and barcodes into the read name column but also place all reads in different subfolders based on their sequencing quality and sequence complexity. Then, we propose a distributed suffix tree algorithm with supervised graph mining on Spark to mitigate the influence of improper string graph construction due to sequencing errors. Using an outlier detection algorithm on a suffix tree, the process in Fig. 10(b) can be configured as a highly sensitive error detector for low-coverage regions. After that, we are able to adopt the parallel string graph construction shown in Fig. 10(c) to represent the relation of each qualified read and the read overlaps by suffix-prefix information. More details are available in our previous paper [25]. Based on the string graph, we propose the parallel haplotype-sensitive assembly (HSA) depicted in Fig. 10(d) to construct haplotype-resolved contigs; the detailed procedure of this module will be described in the next session. For example, these generated contigs (Fig. 10) include (i) heterozygous SNPs, (ii) heterozygous deletions, (iii) heterozygous insertions and (iv) homozygous SNPs. In summary, ConnectedReads transforms noisy short reads with low quality and sequencing errors to long and qualified contigs with clear haplotype information.

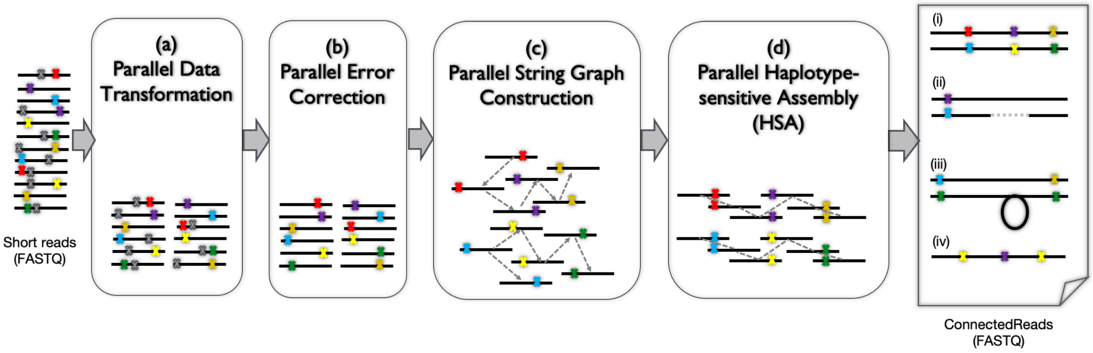

Short reads (FASTQ)
(a) Parallel Data Transformation
(b) Parallel Error Correction
(c) Parallel String Graph Construction
(d) Parallel Haplotype-sensitive Assembly (HSA)
(i)
(ii)
(iii)
(iv)
ConnectedReads (FASTQ)

Fig. 10. Workflow of ConnectedReads. (a) Parallel data transformation. (b) Parallel error correction. (c) Parallel string graph construction. (d) Parallel haplotype-sensitive assembly (HSA).

Parallel HSA

ConnectedReads adopts a string graph, a lossless data representation that is fundamental for many de novo assemblers based on the overlap-layout-consensus paradigm [26, 27], to represent the overlaps of each read. Here, we propose a Spark-based HSA based on a string graph. By leveraging GraphFrame [28], HSA is able to perform efficient and scalable graph traversal operations and supervised graph mining algorithms on Apache Spark. Before introducing the detailed algorithms of HSA, we formally define the notations of the sequencing data and string graph that we will use in the following sections.

Definitions and notation

Let $G(V, E)$ be a directed graph and $S, T \subseteq V$. We define $V = \{v_1, v_2, \ldots, v_k\}$ and $E(S, T)$ to be the set of edges going from S to T, i.e.,

$$|V| = k \text{ and } E(S, T) = \{\, e_{ij} \in E : v_i \in S,\, v_j \in T \text{ and } i < j \leq k \,\}.$$

Definition 1. Let R be the set of short reads and R^{RC} be the reverse complement set of R. G and G^{RC} are the string graphs based on R and R^{RC}, respectively. If e_{ij} exists in G and $i, j \in R$, then e_{mn} also exists in G^{RC} and $m,n \in R^{RC}$ such that m and n are the reverse complements of j and i, respectively.

We define $G^{EXT}(V, E)$ to be the expanded graph of vertices $V \in R \cup R^{RC}$ and edges E to be the set of edges in either G or G^{RC}.

Definition 2. Let *IN(v_i)* and *OUT(v_i)* be the number of in-degree and out-degree edges of v_i, respectively. We define a vertex v_i to be

Singleton if *IN(v_i)* = 0 and *OUT(v_i)* = 0;
Start(S) if *IN(v_i)* = 0 and *OUT(v_i)* = 1;
Terminator(T) if *IN(v_i)* = 1 and *OUT(v_i)* = 0;
Bridge(B) if *IN(v_i)* = 1 and *OUT(v_i)* = 1;
and **Ambiguity(A)** if *IN(v_i)* ≠ 1 or *OUT(v_i)* ≠ 1.

Definition 3. We define the priority order of the vertex labels to be **T** > **S** > **B** and **A**.

This means that **B** could be relabeled as **S** or **T**. Once a vertex becomes **T**, it will always be **T** regardless of what the graph property propagation is.

To keep the depth information in FASTQ format, a new quality encoding function is proposed.

Definition 4. Let *L* and *N* be the numbers of layers for quality and depth, respectively. *L**N* should be 42 if Phred33 is adopted. Let *Q[i]* and *D[i]* be the quality and depth of the *i*-th base of the given contig, respectively. We define the quality encoding function $Encoder_Q[i]$ and the depth encoding function $Encoder_D[i]$ to be

$$Encoder_Q[i] = \left\lfloor \frac{Q[i] - ord("!")}{N} \right\rfloor * N$$

$$Encoder_D[i] = \begin{cases} 0 & if\ D[i] == 1 \\ 1 & if\ 2 \le D[i] < 4 \\ 2 & if\ 4 \le D[i] < 7 \\ 3 & if\ 7 \le D[i] < 11 \\ & \cdots \\ N-1 & if\ D[i] \ge 200 \end{cases}$$

In addition, we define the quality-depth (QD)-encoding function $Encoder_{QD}[i]$ to be

$$Encoder_{QD}[i] = chr(Encoder_Q[i] + Encoder_D[i] + ord("!"))$$

System flow

In Fig. 11, HSA makes use of five modules, namely, (a) string graph expansion, (b) vertex connectivity classification, (c) the Start-Bridge-Terminator (S-B-T) traversal algorithm, (d) the ploidy-based routing algorithm and (e) the triplet traversal algorithm. To simplify the following graph traversal algorithms, each vertex and its reverse complement should be separated in a string graph. This means that the string graph *G* generated by Fig. 10(c) should first be expanded by Definition 1 and named G^{EXT}, which contains *G* and G^{RC}. For that reason, Fig. 11(a) generates the expanded string graph with all reads and their reverse complements. After removing all singletons in G^{EXT}, the remaining vertices will be classified by the vertex connectivity classification module shown in Fig. 11(b) into four classes, which are defined in Definition 2: start (**S**), bridge (**B**), terminator (**T**) and ambiguity (**A**). To generate haplotype-sensitive contigs, the graph traversal/pairing algorithms shown in Fig. 11(c-e) are proposed based on the above properties of vertices and described comprehensively in the next section.

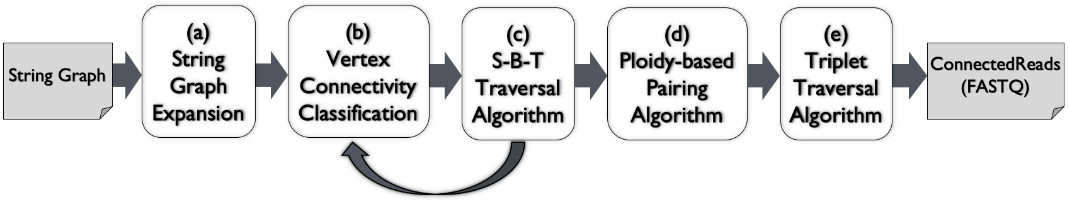

String Graph
(a) String Graph Expansion
(b) Vertex Connectivity Classification
(c) S-B-T Traversal Algorithm
(d) Ploidy-based Pairing Algorithm
(e) Triplet Traversal Algorithm
ConnectedReads (FASTQ)

Fig. 11. Parallel haplotype-sensitive assembly (HSA). (a) String graph expansion. (b) Vertex connectivity classification. (c) S-B-T traversal algorithm. (d) Ploidy-based paired algorithm. (e) Triplet traversal algorithm.

Graph mining algorithms

ConnectedReads leverages Apache Spark and its derived packages to propose three efficient and scalable graph algorithms for large graphical datasets, i.e., string graphs. Spark GraphFrame is a powerful tool for performing distributed computations with large graphical datasets. In addition, Spark Dataset is a type-safe interface that provides the benefits of resilient distributed datasets (RDDs) and Spark SQL optimization. By leveraging GraphFrame and Dataset, we propose several Spark-based graph traversal/routing algorithms for HSA with high performance and scalability.

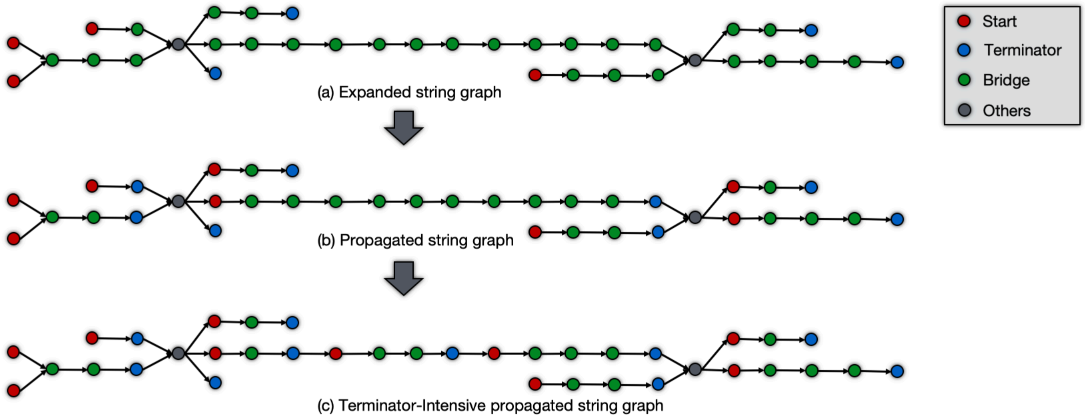

Start
Terminator
Bridge
Others
(a) Expanded string graph
(b) Propagated string graph
(c) Terminator-Intensive propagated string graph

Fig. 12. An example to demonstrate the graph preprocessing workflow for the S-B-T traversal algorithm. (a) Expanded string graph. (b) Propagated string graph. (c) Terminator-intensive propagated string graph.

From our observations of NGS sequencing data, two challenges must be overcome if we want to enhance the performance of graph operation for HSA. The first challenge in the string graph involves long diameters, and the second challenge is how to properly connect the vertices with multiple in-/out-degrees. Taking NA12878 as an example, more than 90% of the vertices are **B,** and the longest diameter from **S** to **T** is more than 500. This means that the traversal algorithm must perform the propagation operation in at least 500 iterations from one vertex (**S**) to another vertex (**T**). For most graph frameworks, the performance of a graph algorithm is strongly related to the number of iterations for its traversal operation. Here, we propose the S-B-T traversal algorithm, shown in Fig. 11(c), to connect all vertices from **S** to **T** via all adjacent **B**s. By following Definition 3, the data preprocessing flow for the expanded string graph is applied, as illustrated in Fig. 12. First, all vertices before and after any vertex **A** should be relabeled as **T** and **S**, respectively. To overcome the long-diameter problem, the mechanism used to relabel **B** to **T** by customized random selection is applied to shorten the diameter of the given graph, and then, the terminator-intensive propagated string graph is acquired. Based on the terminator-intensive graph, the S-B-T traversal algorithm, which integrates a belief propagation algorithm with iterative graph merging, is able to theoretically reduce the time complexity from O(N) to O($\sqrt{N}$) (N: the number of iterations for graph propagation).

When each simple routing path (i.e., **S** to **T** via **B**s) is completely traversed and merged into a single vertex in the new graph, the majority of the vertices have multiple in-/out-degrees. Therefore, we have to solve the second challenge - how to select a proper routing path in the multiple-in-/out-degree graph. Many useful indicators enable us to perform correct routing, such as read pairs, barcodes and read overlaps. The pseudo code of the ploidy-based pairing algorithm shown in Fig. 11(d) is as follows:

```
def ploidy_based_pairing(G: GraphFrame for string graph, N: int for ploidy) {
  Candidates = ∅
  V_a = all of As in G
  for each v ∈ V_a {
    I_v = the set of vertices point to v
    O_v = the set of vertices pointed by v
    D_v = de_noise(I_v, v)
    T = ∅
    for each u ∈ D_v {
      (t, s) = find_best_matching(u, v, O_v)          #t: triple ; s: score
      if s ≥ MIN_THRESHOLD then
        add (t, s) into T
    }
    sort T by score
    add the top N triples from T into Candidates
  }
  return Candidates
}
```

To mitigate false assembly due to sequencing errors, the function **de_noise()** is used to remove the noisy vertices in I_v by using a naïve clustering algorithm based on the read-pair information in this paper. In addition, the function **find_best_matching()** adopts a tripartite clustering algorithm based primarily on the number of supports from read-pair information in (u, v), (u, O_v) and (v, O_v) to find the best combination for sequence assembly. Using the ploidy-based pairing algorithm, we obtain sufficient information to overcome the second challenge and then apply the triplet traversal algorithm shown in Fig. 11(e) to construct the haplotype-sensitive contigs by traversing the aggregated graph from each **S** to **T** via the triple set generated by the ploidy-based pairing algorithm. The triplet traversal algorithm involves almost the same procedures as the S-B-T traversal algorithm, except for the linkage of vertexes. In summary, HSA leverages Apache Spark to efficiently generate haplotype-sensitive contigs from a large string graph.

Performance

Here, sample NA12878 is processed by ConnectedReads on our Spark cluster. It takes approximately 18 hours for the 30X WGS sample; the detailed performance of ConnectedReads is described in Table 6.

Table 6. Execution time for NA12878

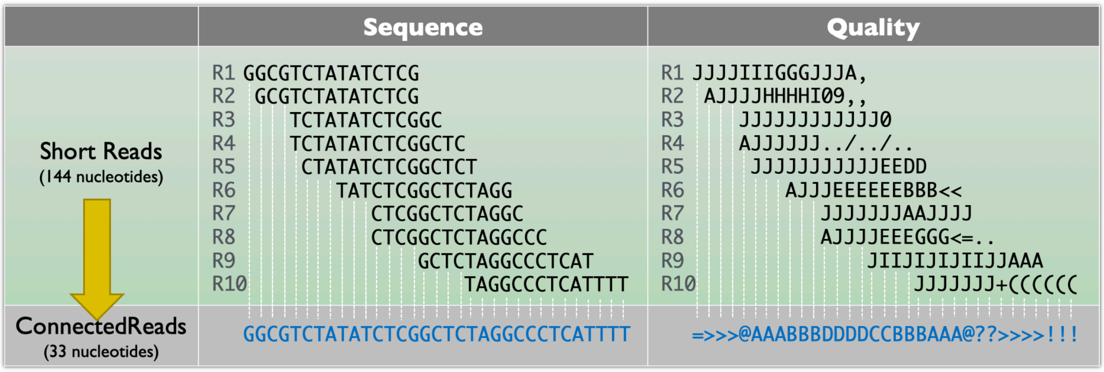
Module	Time (hours)	# Executors	# Cores per executor
(a) Parallel data transformation	1.9	9	17
(b) Parallel error correction	8.1	15	8
(c) Parallel string graph construction	4.3	9	17
(d) Parallel haplotype-sensitive assembly	4.0	9	17
Data export from HDFS to local disk	0.05	9	4
Total	18.3		

*There are 9 computing nodes (each node has two E502650 v4 with 512 GB of memory)

QD encoder/decoder

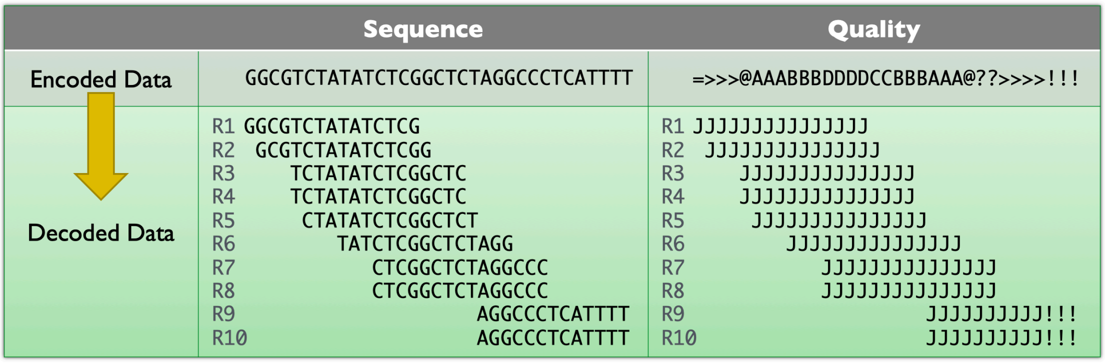
Since ConnectedReads not only connects reads with their overlaps but also aggregates identical reads, the size of ConnectedReads contigs will be reduced by at least 90% in comparison with that of contigs from short-read data. However, the downstream analysis tools might not work well due to the loss of depth information. We have to take the depth information into consideration and keep the output compatible with FASTQ format. Therefore, the QD encoder and decoder are proposed based on Definition 4 to mitigate the impact on information loss. The mechanism of the QD-encoding function is quite flexible, allowing it to fit most use cases. For example, if the depth information is more critical than quality, (L, N) = (3, 14) is the best choice. Fig. 13 provides an example to demonstrate how the QD encoder transforms short reads into ConnectedReads contigs. The data reduction rate is approximately 77% (from

144 characters to 33 characters in both sequence and quality). In addition, Fig. 14 shows the result of applying the QD-decoding function. The recovery rate in terms of sequences and quality is 95.8% (138/144) and 47.3% (71/144), respectively. Therefore, ConnectedReads provides an efficient QD-decoding tool for some use cases that heavily leverages depth information.



	Sequence	Quality
Short Reads (144 nucleotides)	R1 GGCGTCTATATCTCG	R1 JJJJIIIGGGJJJA,
	R2 GCGTCTATATCTCG	R2 AJJJJHHHHI09,,
	R3 TCTATATCTCGGC	R3 JJJJJJJJJJJJ0
	R4 TCTATATCTCGGCTC	R4 AJJJJJJ../../..
	R5 CTATATCTCGGCTCT	R5 JJJJJJJJJJJJEEDD
	R6 TATCTCGGCTCTAGG	R6 AJJJEEEEEEBBB<<
	R7 CTCGGCTCTAGGC	R7 JJJJJJJAAJJJJ
	R8 CTCGGCTCTAGGCCC	R8 AJJJJEEEGGG<=..
	R9 GCTCTAGGCCCTCAT	R9 JIIJIJIJIIJJAAA
	R10 TAGGCCCTCATTTT	R10 JJJJJJJ+((((((
ConnectedReads (33 nucleotides)	GGCGTCTATATCTCGGCTCTAGGCCCTCATTTT	=>>>@AAABBBDDDDCCBBBAAA@??>>>>!!!

Fig. 13. An example application of the QD encoder (taking (L, N) = (3, 14) as an example).



	Sequence	Quality
Encoded Data	GGCGTCTATATCTCGGCTCTAGGCCCTCATTTT	=>>>@AAABBBDDDDCCBBBAAA@??>>>>!!!
Decoded Data	R1 GGCGTCTATATCTCG	R1 JJJJJJJJJJJJJJJ
	R2 GCGTCTATATCTCGG	R2 JJJJJJJJJJJJJJJ
	R3 TCTATATCTCGGCTC	R3 JJJJJJJJJJJJJJJ
	R4 TCTATATCTCGGCTC	R4 JJJJJJJJJJJJJJJ
	R5 CTATATCTCGGCTCT	R5 JJJJJJJJJJJJJJJ
	R6 TATCTCGGCTCTAGG	R6 JJJJJJJJJJJJJJJ
	R7 CTCGGCTCTAGGCCC	R7 JJJJJJJJJJJJJJJ
	R8 CTCGGCTCTAGGCCC	R8 JJJJJJJJJJJJJJJ
	R9 AGGCCCTCATTTT	R9 JJJJJJJJJJ!!!
	R10 AGGCCCTCATTTT	R10 JJJJJJJJJJ!!!

Fig. 14. An example application of the QD decoder (taking (L, N) = (3, 14) and read length = 15 as an example).

Availability of data and materials

Data: The raw sequencing data discussed in this manuscript are deposited on the European Bioinformatics Institute (EBI) and NCBI websites. NA12878 is available from https://www.ebi.ac.uk/ena/data/view/ERR2438055; NA24694 and NA19240 are available from https://www.ncbi.nlm.nih.gov/sra/SRX1388455 and https://www.ncbi.nlm.nih.gov/sra/SRX4637790, respectively.

Codes: The source code and scripts for the ConnectedReads workflow and the related experiments discussed in this paper are available at https://github.com/atgenomix/connectedreads.

Abbreviations

CNV: Copy number variant

HDFS: Hadoop distributed file system

HSA: Haplotype-sensitive assembly

LRS: Long-read sequencing

NGS: Next-generation sequencing

NRNR: Non-reference, non-repetitive

QD: Quality depth

RDD: Resilient distributed dataset

SRS: Short-read sequencing

SV: Structural variant

UNI: Unique non-reference insertion

WGS: Whole-genome sequencing

References

1. Esplin, E.D., L. Oei, and M.P. Snyder, *Personalized sequencing and the future of medicine: discovery, diagnosis and defeat of disease.* Pharmacogenomics, 2014. **15**(14): p. 1771-1790.
2. Ku, C.S., et al., *Integrating next-generation sequencing into the diagnostic testing of inherited cancer predisposition.* Clin Genet, 2013. **83**(1): p. 2-6.
3. Pfeifer, S.P., *From next-generation resequencing reads to a high-quality variant data set.* Heredity (Edinb), 2017. **118**(2): p. 111-124.
4. Li, H., *Toward better understanding of artifacts in variant calling from high-coverage samples.* Bioinformatics, 2014. **30**(20): p. 2843-51.
5. Li, H., *Exploring single-sample SNP and INDEL calling with whole-genome de novo assembly.* Bioinformatics, 2012. **28**(14): p. 1838-44.
6. Gupta, S.e.a. *SPARK: A high-level synthesis framework for applying parallelizing compiler transformations.* in *VLSI Design.* 2003. 16th International Conference on IEEE.

7. Parikh, H., et al., *svclassify: a method to establish benchmark structural variant calls.* BMC Genomics, 2016. **17**: p. 64.

8. Wong, K.H.Y., M. Levy-Sakin, and P.Y. Kwok, *De novo human genome assemblies reveal spectrum of alternative haplotypes in diverse populations.* Nat Commun, 2018. **9**(1): p. 3040.

9. Kehr, B., et al., *Diversity in non-repetitive human sequences not found in the reference genome.* Nat Genet, 2017. **49**(4): p. 588-593.

10. Hatem, A., et al., *Benchmarking short sequence mapping tools.* BMC Bioinformatics, 2013. **14**: p. 184.

11. *GATK Best Practice*. Available from: https://software.broadinstitute.org/gatk/best-practices/.

12. Li, H., *Minimap2: pairwise alignment for nucleotide sequences.* Bioinformatics, 2018. **34**(18): p. 3094-3100.

13. Cameron, D.L., L. Di Stefano, and A.T. Papenfuss, *Comprehensive evaluation and characterisation of short read general-purpose structural variant calling software.* Nat Commun, 2019. **10**(1): p. 3240.

14. *Nanopore*. Available from: https://nanoporetech.com/.

15. *PacBio*. Available from: https://www.pacb.com/.

16. Shi, L., et al., *Long-read sequencing and de novo assembly of a Chinese genome.* Nat Commun, 2016. **7**: p. 12065.

17. Merker, J.D., et al., *Long-read genome sequencing identifies causal structural variation in a Mendelian disease.* Genet Med, 2018. **20**(1): p. 159-163.

18. Sudmant, P.H., et al., *An integrated map of structural variation in 2,504 human genomes.* Nature, 2015. **526**(7571): p. 75-81.

19. Molnar, J., et al., *The genome of the chicken DT40 bursal lymphoma cell line.* G3 (Bethesda), 2014. **4**(11): p. 2231-40.

20. Deininger, P., *Alu elements: know the SINEs.* Genome Biol, 2011. **12**(12): p. 236.

21. *pbsv*. Available from: https://github.com/pacificbiosciences/pbsv/.

22. Li, H., *FermiKit: assembly-based variant calling for Illumina resequencing data.* Bioinformatics, 2015. **31**(22): p. 3694-6.

23. *UCSC LiftOver*. Available from: https://genome.ucsc.edu/cgi-bin/hgLiftOver.

24. Massie, M.a.N., Frank and Hartl, Christopher and Kozanitis, Christos and Schumacher, Andre and Joseph, Anthony D and Patterson, David A, *ADAM: Genomics Formats and Processing Patterns for Cloud Scale Computing*, in

UCB/EECS-2013-207. 2013: EECS Department, University of California, Berkeley.

25. Chung-Tsai Su, M.-T.C., Yun-Chian Cheng, Yun-Lung Li, Yao-Ting Wang, *GraphSeq: Accelerating String Graph Construction for De Novo Assembly on Spark.* 2016.
26. Simpson, J.T. and R. Durbin, *Efficient de novo assembly of large genomes using compressed data structures.* Genome Res, 2012. **22**(3): p. 549-56.
27. Bonizzoni, P., et al., *FSG: Fast String Graph Construction for De Novo Assembly.* J Comput Biol, 2017. **24**(10): p. 953-968.
28. *GraphFrame.* Available from: https://github.com/graphframes/graphframes.

Acknowledgements

We are grateful to Prof. Chen Chia-Hsiang and his team for supporting the Sanger sequencing validation. In addition, Cheng Yun-Chian and Prof. Chen Chien-Yu assisted in algorithm design in the early stage. The results presented in the paper are based on the data generated by the Atgenomix Seqslab Cloud platform.

Funding

The research was supported by Atgenomix Inc.

Author information

Affiliations

Atgenomix Inc., Taiwan, R.O.C.

Chung-Tsai Su, Sid Weng, Yun-Lung Li & Ming-Tai Chang

Contributions

CS, SW, YL and MC developed the algorithms and implemented the tools for data transformation and error correction. CS developed the algorithms and implemented the tools for string graph construction. CS and SW developed the algorithms and implemented the tools for haplotype-sensitive assembly. CS collected the sequencing data and carried out the benchmarks based on the sequencing dataset. CS carried out the experiments, developed the structural variant caller and carried out the related experiments. CS prepared the manuscript with input from all other authors. All authors read and approved the final manuscript.

Corresponding author

Correspondence to Chung-Tsai Su

Ethics declarations

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

CS, SW, YL and MC are employees of Atgenomix Inc. In addition, they all hold shares, stock options or restricted stock units in Atgenomix Inc.

Additional information

Additional file 1: The detailed information for each table and figure. (XLSX)
Additional file 2: The FASTA file for insertions. (TEXT)
Additional file 3: The FASTA file for soft-clipped sequences. (TEXT)
Additional file 4: The FASTA file for unmapped contigs. (TEXT)
Additional file 5: Sanger Validation for N-Gap cases. (DOCX)
Additional file 6: Review history. (DOCX)