Informatics for PacBio Long Reads

Yuta Suzuki

Abstract In this article, we review the development of a wide variety of bioinformatics software implementing state-of-the-art algorithms since the introduction of SMRT sequencing technology into the field. We focus on the three major categories of development: read mapping (aligning to reference genomes), *de novo* assembly, and detection of structural variants. The long SMRT reads benefit all the applications, but they are achievable only through considering the nature of the long reads technology properly.

Advances in SMRT Biology and Challenges in Long Read Informatics

In 2011, advent of the PacBio RS sequencer and its SMRT (single molecule real-time) sequencing technology revolutionized the concept of DNA sequencing. Longer reads are promised to generate de novo assembly of much higher contiguity, and the claim was proved by several assembly projects (Steinberg et al. 2014; Pendleton et al. 2015; Seo et al. 2016). The lack of sequencing bias was proved to be able to read regions which are extremely difficult for NGS (Next Generation Sequencers) (Loomis et al. 2013).

None of these achievement, however, was just straightforward application of conventional informatics strategy developed for short read sequencers; the virtue of the long reads was not free at all. As many careful skeptics claimed in the early history of PacBio sequencing, the long reads seemed too noisy. Base accuracy was around ~85% for single raw read, that is, ~15% of bases were wrong calls, and

Y. Suzuki (✉)
Department of Computational Biology and Medical Sciences, Graduate School of Frontier Sciences, The University of Tokyo, Tokyo, Japan
e-mail: yuta_suzuki@edu.k.u-tokyo.ac.jp

Y. Suzuki (ed.), *Single Molecule and Single Cell Sequencing*, Advances in Experimental Medicine and Biology 1129,
https://doi.org/10.1007/978-981-13-6037-4_8

indels consisted most of the errors. The higher error rate made it inappropriate to apply informatics tools designed for much accurate short read technologies.

Even the higher error rate is properly handled by sophisticated algorithms, the length of the reads itself can pose another problem. Computational burden of many algorithms depends on the read length L. When only the short reads are assumed, it may be considered as constant, e.g., L = 76, 150, etc. The emergence of long read sequencer changed the situation drastically by improving the read length by orders of magnitudes, to thousands of bases, and to tens of thousands of bases by now. Besides the ongoing innovations for longer reads, there is a large variation in length of sequencing reads even in the same sequencing run. Therefore, the assumption that the read length is constant is not valid anymore, and one must have a strategy to handle (variably) long reads in reduced time (CPU hours) and space (memory footprint) requirement.

Availability of long read opened a door to the set of problems which were biologically existing in real but implicitly ignored by studies using short read sequencing. For example, we had to realize that a non-negligible fraction of reads could cover SVs (structural variants), requiring a new robust mapping strategy other than simply masking the known repetitive regions.

Consequently, many sophisticated algorithms had to be developed to resolve these issues; how to mitigate higher error rate, and how it can be done efficiently for long reads. The rest of this article covers some important innovations achieved and ongoing efforts in informatics area to make the most of long reads data.

Aligning Noisy Long Reads with Reference Genome

When one aligns long reads against reference sequence, one must be aware that the variations between reads and reference stems from two conceptually separate causes. On one hand, there are sequencing errors in its simple sense, which is discrepancy between a read observed and actual sequence being sequenced. On the other hand, we expect a sample sequenced would have slightly different sequence than a reference sequence (otherwise there is no point in doing sequencing), and those difference are usually called variants. Though sequencing errors and sequence variants are conceptually different, however, they both appears just as “errors” to us unless they have some criteria to distinguish them. The next two examples are for understanding why the distinction between two classes of “error” is relevant here.

Let’s consider we have some *noisy* reads. Clearly, we cannot call sequence variants specific to the sample unless the frequency of sequencing errors is controlled to be sufficiently low compared to the frequency of variants. This is the reason why it is difficult for noisy reads to detect small nucleotide variants such as point mutations and indels.

Next, assume we have *long* reads. Then, there are more chances that the reads span the large variations such as structural variations (SVs) between a reference genome and the sample sequenced. This situation is problematic for aligners who

considered any possible variation between reads and reference to be sequencing errors, for such aligners would fail to detect correct alignment as they need to introduce too much errors for aligning these sequences. Some aligners try to combat the situation by employing techniques such as chaining and split alignment. Some aligners (NGMLR, Minimap2) explicitly introduce an SV-aware scoring scheme such as a two-parts concave gap penalty, which reflects the two classes of variations between read and reference.

Sequence alignment is so fundamental in sequence analysis that it finds its application everywhere. For example, mapping sequencing reads to reference genome is the very first step of resequencing studies. Accuracy of mapping can directly be translated into the overall reliability of results. Also, mapping is often one of the most computationally intensive steps. Therefore, accurate and faster mapping software would benefit the whole area of resequencing studies. In the context of de novo assembly pipeline, it is used for detecting overlap among long reads. Of noted, desired balance of sensitivity and specificity of overlap detection is controlled differently from mapping to reference, and it could often be very subtle.

Though it is more or less subjective to make distinction between standalone aligners and aligners designed as a module of assembly pipeline or SV detection pipeline, we decided to cover some aligners in other sections. MHAP will be introduced in relation with Canu in the section devoted to assembly tools. Similarly, NGMLR will be detailed together with Sniffle in the section for SV detection.

BWA-SW and BWA-MEM

Adopting the seed-and-extend approach, BWA-SW (Li & Durbin 2010) builds FM-indices for both query and reference sequence. Then, DP (dynamic programming) is applied to these FM-indices to find all local matches, i.e., seeds, allowing mismatches and gaps between query and reference. Detected seeds are extended by Smith-Waterman algorithm. Some heuristics are explicitly introduced to speed up alignment of large-scale sequencing data and to mitigate the effect of repetitive sequences. BWA-MEM (Li 2013) inherits similar features implemented in BWA-SW such as split alignment, but is found on a different seeding strategy using SMEM (supermaximal exact matches) and *reseeding* technique to reduce mismapping caused by missing seed hits.

BLASR

BLASR (Chaisson & Tesler 2012) (Basic Local Alignment with Successive Refinement) is also one of the earliest mapping tools specifically developed for SMRT reads. Like BWA-MEM, it is probably the most widely used one to date. Bundled with official SMRT Analysis, it has been the default choice for the

mapping (overlapping) step in all protocols such as resequencing, de novo assembly, transcriptome analysis, and methylation analysis. In the BLASR's paper, the authors explicitly stated it was designed to combine algorithmic devices developed in two separate lines of studies, namely, a coarse alignment method for whole genome alignment and a sophisticated data structure for fast short read mapping. Proven to be effective for handling noisy long read, the approach of successive refinement, or seed-chain-align paradigm, has become a standard principle.

BLASR first finds short exact matches (anchors) using either suffix array or FM index (Ferragina & Manzini 2000). Then, the regions with clustered anchors aligned colinearly are identified as candidate mapping locations, by global chaining algorithm (Abouelhoda & Ohlebusch 2003). The anchors are further chained by sparse dynamic programming (SDP) within each candidate region (Eppstein et al. 1992). Finally, it gives detailed alignment using banded DP (dynamic programming) guided by the result of SDP. BLASR achieved tenfold faster mapping of reads to human genome than BWA-SW algorithm at comparable mapping accuracy and memory footprint.

DALIGNER

DALIGNER (Myers 2014) is specifically designed for finding overlaps between noisy long reads, though its concept can also be adopted for a generic long read aligner, as implemented in DAMAPPER (https://github.com/thegenemyers/DAMAPPER). Like in BLASR, DALIGNER also performs filter based on short exact matches. Instead of using BWT (FM index), it explicitly processes k-mers within reads by thread-able and cache coherent implementation of radix sort. Detected k-mers are then compared via block-wise merge sort, which reduces memory footprint to a constant depending only on the block size. To generate local alignment, it applies O(ND) diff algorithm between two candidate reads (Myers 1986). DALIGNER achieved 22 ~ 39-fold speedup over BLASR at higher sensitivity in detecting correct overlaps (Myers 2014). DALIGNER is supposed to be a component for read overlap (with DAMASKER for repeat masking, DASCRUBBER for cleaning up low quality regions, and a core module for assembly) of DAZZLER de novo assembler for long noisy reads, which will be released in future.

Minimap2

Minimap2 (Li 2017) is one of the latest and state-of-the-art alignment program. Minimap2 is general-purpose aligner in that it can align short reads, noisy long reads, and reads from transcripts (cDNA) back to a reference genome. Minimap2 combines several algorithmic ideas developed in the field, such as locality-sensitive

hashing as in Minimap and MHAP. For accounting possible SVs between reads and genome, it employs concave gap cost as in NGMLR, and it is efficiently computed using formulation proposed by Suzuki & Kasahara (2017). In addition to these features, the authors further optimized the algorithm, by transforming the DP matrix from row-column coordinate to diagonal-antidiagonal coordinate for better concurrency in modern processors. According to the author of Minimap2, it is supposed to replace BWA-MEM, which is in turn a widely used extension of BWA-SW.

De novo Assembly

As Lander-Waterman theory (Lander & Waterman 1988) would assert, the longer input reads are quite essential in achieving a high-quality genome assembly for repetitive genomes. Therefore, developing a *de novo* assembler for long read is naturally the most active area in the field of long read informatics.

To our knowledge, almost all assemblers published for long read take an overlap-layout-consensus (OLC) approach, where the overall task of assembly can be divided into the three steps. (1. Overlap) The overlaps between reads are identified as candidate pairs representing the same genomic regions, and the overlap graph is constructed to express these relations. (2. Layout) The graph is transformed to generate linear contigs. The step often starts by constructing the string graph (Myers 2005), a string-labeled graph which encodes all the information in reads observed, and eliminates edges containing redundant information. (3. Consensus) The final assembly is polished. To eliminate errors in contigs, consensus is taken among reads making up the contigs.

Though we do not cover tools for the consensus step here, there are many of them released to date including official Quiver and Arrow bundled in SMRT Analysis (https://github.com/PacificBiosciences/GenomicConsensus), another official tool pbdagcon (https://github.com/PacificBiosciences/pbdagcon), Racon (Vaser et al. 2017), and MECAT (Xiao et al. 2017). Of note, quality of a polished assembly can be much better than a short-read-based assembly due to the randomness of sequencing errors in long reads (Chin et al. 2013; Myers 2014).

FALCON

FALCON (Chin et al. 2016) is designed as a diploid-aware *de novo* assembler for long read. It starts by carefully taking consensus among the reads to eliminate sequencing errors while retaining heterozygous variants which can distinguish two homologous chromosomes (FALCON-sense). For constructing a string graph, FALCON runs DALIGNER. The resulted graph contains "haplotype-fused" contigs and "bubbles" reflecting variations between two homologous chromosomes. Finally, FALCON-unzip tries to resolve such regions by phasing the associated long reads

and local re-assembly. The contigs obtained are called “haplotigs”, which are supposed to be faithful representation of individual alleles in the diploid genome.

Canu (& MHAP)

MHAP (Berlin et al. 2015) (Min-Hash Alignment Process) utilized MinHash for efficient dimensionality reduction of the read space. In MinHash, H hash functions are randomly selected, each of them maps k-mer into an integer. For a given read of length L, only the minimum values over the read are recorded for each of H hash functions. The k-mers at which the minimum is attained are called *min-mers*, and resulted representation is called a *sketch*. The sketch serves as a locality sensitive hashing of each read, for the similar sequences are expected share similar sketches. Because the sketch retains the data only on H min-mers, its size is fixed to H, independent of read length L.

Built on top of MHAP, Canu (Koren et al. 2017) extends best overlap graph (BOG) algorithm (Miller et al. 2008) for generating contigs. A new “bogart” algorithm estimates an optimal overlap error rate instead of using predetermined one as in original BOG algorithm. This requires multiple rounds of read and overlap error correction, but eventually enables to separate repeats diverged only by 3%. Though BOG algorithm is greedy, the effect is mitigated in Canu by inspecting non-best overlaps as well to avoid potential misassemblies.

HINGE

While there is no doubt that obtaining more contiguous (i.e., higher contig N50) assembly is a major goal in genome assembly, the quest just for longer N50 may cause misassemblies if the strategy gets too greedy. Being aware that danger, HINGE (Kamath et al. 2017) aims to perform the optimal resolution of repeats in assembly, in the sense that the repeats should be resolved if and only if it is supported by long read data available. To implement such a strategy is rather straightforward for de Bruijn graphs. In de Bruijn graph, its k-mers representing nodes are connected by edges when they co-occur next to each other in reads. In ideal situation, the genome assembly is realized as an Eulerian path, i.e., trail which visits every edge exactly once, in the de Bruijn graph. However, de Bruijn graphs are not robust for noisy long read, so overlap graphs are usually preferred for long read. One of the key motivations of HINGE is to give such a desirable property of de Bruijn graphs, to overlap graphs which is more error-resilient. To do so, HINGE enriches string graph with additional information called “hinges” based on the result of the read overlap step. Then, assembly graph with optimal repeat resolution can be constructed via a hinge-aided greedy algorithm.

Miniasm (& Minimap)

Minimap (Li 2016) adopts a similar idea as MHAP, it uses *minimizers* to represents the reads compactly. For example, Minimap uses a concept of (w,k)-minimizer, which is the smallest (in the hashed value) k-mer in w consecutive k-mers. To perform mapping, Minimap searches for colinear sets of minimizers shared between sequences. Miniasm (Li 2016), an associated assembly module, generates assembly graph without error-correction. It firstly filters low-quality reads (chimeric or with untrimmed adapters), constructs graph greedily, and then cleans up the graph by several heuristics, such as popping small bubbles and removing shorter overlaps.

Detection of Structural Variants (SVs)

Sequence variants are called *structural* when they are explained by the mechanisms involving double-strand breaks, and are often defined to be variants larger than certain size (e.g., 50 bp) for the sake of convenience. They are categorized into several classes such as insertions/deletions (including presence/absence of transposons), inversion, (segmental) duplication, tandem repeat expansion/contraction, etc. While some classes of SVs are notoriously difficult to detect via short reads (especially long inversions and insertions), long reads have promise to detect more of them by capturing entire structural events within sequencing reads.

PBHoney

PBHoney (English, Salerno & Reid 2014) implements combination of two methods for detecting SVs via read alignment to reference sequence. Firstly, PBHoney exploits the fact that the alignment of reads by BLASR should be interrupted (giving soft-clipped tails) at the breakpoints of SV events. PBHoney detects such interrupted alignments (piece-alignments) and clusters them to identify individual SV events. Secondly, PBHoney locates SVs by examining the genomic regions with anomalously high error rate. Such a large discordance can signal the presence of SVs because sequencing errors within PacBio reads are supposed to distribute rather randomly.

Sniffles (& NGMLR)

NGMLR (Sedlazeck et al. 2017) is a long-read aligner designed for SV detection, which uses two distinct gap extension penalties for different size range of gaps (i.e., concave gap penalty) to align entire reads over the regions with SVs. Intuitively, the

concave gap penalty is designed so that it can allow longer gaps in alignment while shorter gaps are penalized just as sequencing errors. Adopting such a complicated scoring scheme makes the alignment process computationally intensive (Miller et al. 1988), but NGMLR introduces heuristics to perform faster alignment. Then, an associated tool to detect SVs, Sniffles scans the read alignment to report putative SVs which are then clustered to identify individual events and evaluated by various criteria. Optionally, Sniffle can infer genotypes (homozygous or heterozygous) of detected variants, and can associate "nested SVs" which are supported by the same group of long reads.

SMRT-SV

SMRT-SV (Huddleston et al. 2017) is a SV detection tool based on local assembly. It firstly maps long reads to reference genome, against which SVs are called. Then it searches signatures of SVs within alignment results, and 60 kbp regions around the detected signatures are extracted. The regions are to be assembled locally from those reads using Canu, then SVs are called by examining the alignment between assembled contigs and reference. Local assembly is performed for other regions (without SV signatures) as well to detect smaller variants.

Beyond DNA – Transcriptome Analysis and Methylation Analysis

SMRT sequencing has been found its applications outside DNA analysis as well. When it is applied to cDNA sequencing, long read would be expected to capture the entire structures of transcripts to elucidate expressing isoforms comprehensively. IDP (Isoform Detection and Prediction) (Au et al. 2013) and IDP-ASE (Deonovic et al. 2017) are tools dedicated to analyze long read transcriptome data. To detect expressing isoforms from long read transcriptome data, IDP formulates it in the framework of integer programming. To estimate allele-specific expression both in gene-level and isoform-level, IDP-ASE then solves probabilistic model of observing each allele in short read RNA-seq. Both IDP and IDP-ASE effectively combines long read data for detection of overall structure of transcripts, and short read data for accurate base-pair level information.

In methylation analysis, official *kineticsTools* in SMRT Analysis has been widely used to detect base modification sites and to estimate sequence motives for DNA modification (see (Flusberg et al. 2010) for the principle of detection). Detecting 5-methyl-cytosines (5mC), which is by far the dominant type of DNA modification in plants and animals, is challenging due to their subtle signal. Designed for detecting 5mC modifications in large genomes within practical sequencing depth, AgIn

(Suzuki et al. 2016) exploits the observation that CpG methylation events in vertebrate genomes are correlated over neighboring CpG sites, and tries to assign the binary methylation states to CpG sites based on the kinetic signals under the constraint that a certain number of neighboring CpG sites should be in the same state. Making the most of high mappability of long read, AgIn has been applied to observe diversified CpG methylation statuses of centromeric repeat regions in fish genome (Ichikawa et al. 2017), and to observe allele-specific methylation events in human genomes.

Concluding Remarks

We have briefly described some innovative ideas in bioinformatics for an effective use of long read data. As concluding remarks, let me mention a few prospects for the future development in the field. By now, it is evident the quest for complete genome assembly is almost done, but the remaining is the most difficult part such as extremely huge repeats, centromeres, telomeres. While many state-of-the-art assemblers take the presence of such difficult regions into account and can carefully generate high quality assembly for the rest of genomes, it is remained open how to tackle these difficult part of the genome, how to resolve its sequence, not escaping from them.

Base modification analysis using PacBio sequencers may also have huge potential to distinguish several types of base modifications and to detect them simultaneously in the same sample (Clark et al. 2011), but only the limited number of modification types (6 mA, 4mC, and 5mC) are considered for now. This is mainly due to the technical challenge to alleviate noise in kinetics data to distinguish each type of modifications and unmodified bases from each other.

That said, it will be no doubt that the field would be more attractive than ever, as the use of long read sequencer becomes a daily routine in every area of biological research, or maybe even in clinical practice.

Acknowledgements I'd like to thank Yoshihiko Suzuki, Yuichi Motai and Dr./Prof. Shinichi Morishita for insightful comments on the draft.

References

Abouelhoda MI, Ohlebusch E. A local chaining algorithm and its applications in comparative genomics. International workshop on algorithms in bioinformatics. Berlin/Heidelberg: Springer; 2003.

Au KF, et al. Characterization of the human ESC transcriptome by hybrid sequencing. Proc Natl Acad Sci. 2013;110(50):E4821–30.

Berlin K, et al. Assembling large genomes with single-molecule sequencing and locality-sensitive hashing. Nat Biotechnol. 2015;33(6):623–30.

Chaisson MJ, Tesler G. Mapping single molecule sequencing reads using basic local alignment with successive refinement (BLASR): application and theory. BMC Bioinformatics. 2012;13(1):238.
Chin C-S, et al. Nonhybrid, finished microbial genome assemblies from long-read SMRT sequencing data. Nat Methods. 2013;10(6):563–9.
Chin C-S, et al. Phased diploid genome assembly with single-molecule real-time sequencing. Nat Methods. 2016;13(12):1050–4.
Clark TA, et al. Direct detection and sequencing of damaged DNA bases. Genome Integr. 2011;2(1):10.
Deonovic B, et al. IDP-ASE: haplotyping and quantifying allele-specific expression at the gene and gene isoform level by hybrid sequencing. Nucleic Acids Res. 2017;45(5):e32.
English AC, Salerno WJ, Reid JG. PBHoney: identifying genomic variants via long-read discordance and interrupted mapping. BMC Bioinformatics. 2014;15(1):180.
Eppstein D, et al. Sparse dynamic programming I: linear cost functions. J ACM (JACM). 1992;39(3):519–45.
Ferragina P, Manzini G. Opportunistic data structures with applications. Foundations of computer science, 2000. Proceedings. 41st annual symposium on. IEEE, 2000.
Flusberg BA, et al. Direct detection of DNA methylation during single-molecule, real-time sequencing. Nat Methods. 2010;7(6):461–5.
Huddleston J, et al. Discovery and genotyping of structural variation from long-read haploid genome sequence data. Genome Res. 2017;27(5):677–85.
Ichikawa K, et al. Centromere evolution and CpG methylation during vertebrate speciation. Nat Commun. 2017;8(1):1833.
Kamath GM, et al. HINGE: long-read assembly achieves optimal repeat resolution. Genome Res. 2017;27(5):747–56.
Koren S, et al. Canu: scalable and accurate long-read assembly via adaptive k-mer weighting and repeat separation. Genome Res. 2017;27(5):722–36.
Lander ES, Waterman MS. Genomic mapping by fingerprinting random clones: a mathematical analysis. Genomics. 1988;2(3):231–9.
Li H. Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. arXiv preprint arXiv. 2013:1303.3997.
Li H. Minimap and miniasm: fast mapping and de novo assembly for noisy long sequences. Bioinformatics. 2016;32(14):2103–10.
Li H. Minimap2: versatile pairwise alignment for nucleotide sequences. arXiv. 2017:1708.
Li H, Durbin R. Fast and accurate long-read alignment with Burrows–Wheeler transform. Bioinformatics. 2010;26(5):589–95.
Loomis EW, et al. Sequencing the unsequenceable: expanded CGG-repeat alleles of the fragile X gene. Genome Res. 2013;23(1):121–8.
Miller W, Myers EW. Sequence comparison with concave weighting functions. Bull Math Biol. 1988;50(2):97–120.
Miller JR, et al. Aggressive assembly of pyrosequencing reads with mates. Bioinformatics. 2008;24(24):2818–24.
Myers EW. An O (ND) difference algorithm and its variations. Algorithmica. 1986;1(1):251–66.
Myers EW. The fragment assembly string graph. Bioinformatics. 2005;21(Suppl_2):ii79–85.
Myers G. Efficient local alignment discovery amongst noisy long reads. International workshop on algorithms in bioinformatics. Berlin/Heidelberg: Springer; 2014.
Pendleton M, et al. Assembly and diploid architecture of an individual human genome via single-molecule technologies. Nat Methods. 2015;12(8):780–6.
Sedlazeck FJ, et al. Accurate detection of complex structural variations using single molecule sequencing. bioRxiv. 2017:169557.
Seo J-S, et al. De novo assembly and phasing of a Korean human genome. Nature. 2016;538:243–7.
Steinberg KM, et al. Single haplotype assembly of the human genome from a hydatidiform mole. Genome Res. 2014;24(12):2066–76.

Suzuki H, Kasahara M. Acceleration of nucleotide semi-global alignment with adaptive banded dynamic programming. bioRxiv. 2017:130633.
Suzuki Y, et al. AgIn: measuring the landscape of CpG methylation of individual repetitive elements. Bioinformatics. 2016;32(19):2911–9.
Vaser R, et al. Fast and accurate de novo genome assembly from long uncorrected reads. Genome Res. 2017;27(5):737–46.
Xiao C-L, et al. MECAT: fast mapping, error correction, and de novo assembly for single-molecule sequencing reads. Nat Methods. 2017;14(11):1072–4.