Quantitative Biology 2019, 7(2): 90–109
https://doi.org/10.1007/s40484-019-0166-9

REVIEW

Current challenges and solutions of *de novo* assembly

Xingyu Liao[1], Min Li[1,*], You Zou[1], Fang-Xiang Wu[2], Yi-Pan[3], Jianxin Wang[1,*]

[1] School of Computer Science and Engineering, Central south University, Changsha 410083, China
[2] Division of Biomedical Engineering, University of Saskatchewan, Saskatchewan, S7N 5A9, Canada
[3] Department of Computer Science, Georgia State University, Atlanta, GA 30302, USA
* Correspondence: limin@csu.edu.cn, jxwang@csu.edu.cn

Received April 5, 2018; Revised June 14, 2018; Accepted June 16, 2018

Background: **Next-generation sequencing (NGS) technologies have fostered an unprecedented proliferation of high-throughput sequencing projects and a concomitant development of novel algorithms for the assembly of short reads. However, numerous technical or computational challenges in *de novo* assembly still remain, although many new ideas and solutions have been suggested to tackle the challenges in both experimental and computational settings.**
Results: **In this review, we first briefly introduce some of the major challenges faced by NGS sequence assembly. Then, we analyze the characteristics of various sequencing platforms and their impact on assembly results. After that, we classify *de novo* assemblers according to their frameworks (overlap graph-based, *de Bruijn* graph-based and string graph-based), and introduce the characteristics of each assembly tool and their adaptation scene. Next, we introduce in detail the solutions to the main challenges of *de novo* assembly of next generation sequencing data, single-cell sequencing data and single molecule sequencing data. At last, we discuss the application of SMS long reads in solving problems encountered in NGS assembly.**
Conclusions: **This review not only gives an overview of the latest methods and developments in assembly algorithms, but also provides guidelines to determine the optimal assembly algorithm for a given input sequencing data type.**

Author summary: In this review, we focus on the main challenges facing *de novo* assembly and its solusions. Firstly, we introduce some of the major challenges faced by *de novo* assembly. Secondly, we analyze the characteristics of various sequencing platforms and their impact on assembly results, and introduce the characteristics of each assemblers and their adaptation scene. Thirdly, we introduce in detail the solutions to the main challenges of *de novo* assembly. Finally, we discuss the latest methods and developments in *de novo* assembly.

Keywords: next-generation sequencing; single-cell sequencing; single-molecule sequencing; *de novo* assembly algorithms

INTRODUCTION

De novo genome sequence assembly is the process of reconstructing a genome from a collection of short sequencing reads and is an integral step in any genome project [1,2]. Unlike resequencing projects, *de novo* assembly is performed without the aid of a reference genome; conversely, the genome is reconstructed from scratch. An accurate reconstruction is crucial, as both the continuity and base accuracy of an assembly can affect the results of all downstream analyses [3]. With the increasing efforts to sequence and assemble the genomes of more organisms, the assembly problem becomes more complicated and computationally intensive, especially with short inaccurate sequence reads and genomic repeats [4].

The sequencing errors occur more frequently in regions with an extremely high GC or AT content, such as constant heterochromatin regions, including centromeres, telomere or highly repetitive sequences, all of which may generate a complex assembly graph. In the process of sequence assembly, the complex assembly graph is

difficult to deal with and also the final results are often not satisfactory. The sequencing errors include substitutions, insertions and deletions. The sequencers may output N to indicate the confirmed presence of a nucleotide that cannot be called accurately. The probability of these three types of errors occurring is different with sequencing platforms. Substitution errors are dominant in some platforms such as Illumina, while in others such as 454 and Ion Torrent, homopolymer and carry-forward errors are manifested as plenty of insertions and deletions [5]. The rates and types of sequencing errors vary according to the next-generation platforms and library preparation methods. DNA sequence reads from Illumina sequencing technologies have errors at the rate of 0.02–0.05% [6]. With the diminishing costs, high throughput DNA sequencing has become a commonplace technology in biological research. Whereas the second generation sequencers produces short but quite accurate reads, new technologies such as Pacific Biosciences and Oxford NanoPore produce reads up to 50,000 bp long but with an error rate of at least 15%. Although the long reads have proven to be very helpful in applications like genome assembly [7,8], the error rate poses a challenge for the utilisation of these data.

The sequencing biases occur more frequently in favoring GC-balanced regions and have fewer reads in GC poor regions. For example, Illumina sequencing platform has base composition bias, which usually results in uneven sequencing depth across genome [9]. The base composition bias is usually arisen from different processing steps of Illumina sequencing, such as PCR amplification of library, or sequencing step. Although new experimental technologies, such as optimized PCR protocols, are developed to reduce base bias, the base bias cannot be removed completely in next-generation sequencing (NGS) reads. Since most of *de Bruijn* graph based assemblers use the read depth information for constructing contigs and scaffolds, the uneven sequencing depth impedes the genome assembly [10]. Recent studies have continued to improve assemblies from single cells, but the full potential of single cell sequencing has not yet been realized. The main challenge for single cell assembly is sequencing bias. The challenges faced by single cell genomics are increasing in computation rather than experiment [11]. All previous single cell studies use standard fragment-assembly tools [12,13], developed for data models with characteristics of standard (rather than single cell) sequencing. Sequencing bias poses a serious problem for existing *de novo* assembly algorithms. *De Bruijn*-based assemblers use an average coverage cutoff threshold for contigs to prune out low coverage regions, which tend to include more errors. This pruning step not only reduces the complexity of the underlying *de Bruijn* graph substantially and makes the algorithms practical, but also seriously affect the effective length and genome fraction of the final assembly.

Most genomes contain a certain proportion of repeats, particularly mammalian in which repeats account for 25%–50% of its entire genome [14]. Approximately 50% of the human genome comprises nonrandom repeat elements, such as long interspersed nuclear elements (LINEs), short interspersed nuclear elements (SINEs), long terminal repeats (LTRs) and simple tandem repeats (STRs) [15], which often cause misarrangements or gaps in the assembly. These repeat sequences also cause a nonuniform read depth, thus resulting in copy loss or gain in the assembly. To address the problem of repetitive regions, researchers have proposed some solutions. Most recent assemblers make use of pairedend reads which can be produced by NGS technologies for resolving repetitive region problems. Although paired-end reads are widely applied to resolve problems caused by repetitive regions in genome assembly, the performance of most assemblers is not satisfactory yet because of the existence of sequencing errors and uneven sequencing depths. In contrast, single-molecule sequencing (SMS) long reads that cover the repeats could easily address the problem. Because the SMS long reads are generated by a PCR-free method and are less biased to regions with GC or AT contents [16,17], they are greatly beneficial in overcoming the uneven sequencing depth and gap problems. Alternatively the use of SMS long reads as offered by the PacBio RS methodology can potentially solve complex genomic situations, yet the algorithmic implementation still suffers from a relatively high error rate.

De novo assemblers are often based on the graph structure (such as *de Bruijn* graph or overlap graph) implementation, which require substantial random access memory (RAM), storage and long computation times. For example, several short-read assembly packages (such as Abyss [18], ALLPATHS-LG [19], SGA [20] and SOAPdenovo [21]) have been proven for mammalian-size genomes up to the 3 Gbp human genome, they generally take several days to weeks and require servers or clusters with 512 gigabytes (GB) of RAM and many terabytes (TB) of disk space available for a gigabase-sized genome [22].

Algorithms for *de novo* assembly have evolved in concert with these technology improvements. The main algorithmic approaches to *de novo* assembly are based on a separate theoretical graph framework. There are three basic graph frameworks for efficiently completing their task, namely, overlap layout consensus (OLC) graph, *de Bruijn* graph and string graph. In an OLC graph, overlaps between all reads are first detected, then contigs are formed by iteratively merging overlapping reads until a read is heuristically determined to be at the boundary of a repeat [23]. In a *de Bruijn* graph, nodes are the set of

distinct *k*-mers (substrings of length *k*) extracted from reads and the edges are the (*k*–1) overlap among them. The string graph is a simplified version of a classical overlap graph, where nodes are the sequenced reads and the non-transitive edges encode their suffix-to-prefix overlaps [24–27]. The overlap-based approach is a straight forward approach for long read assembly because it assembles the long reads themselves without converting to *k*-mers [28].

SEQUENCING DATA ANALYSIS

High-throughput sequencing has begun to revolutionize science and healthcare by allowing users to acquire genome wide data by using massively parallel sequencing approaches. The different sequence platform vendors have devised different strategies to prepare the sequence libraries according to suitable templates as well as to detect the signal and ultimately read the DNA sequence. For the Illumina, Solid, PGM and 454 systems, a local clonal amplification of the initial template molecules into polonies [29] is required to increase the signal-to-noise ratio because the systems are not sensitive enough to detect the extension of one base at the individual DNA template molecule level.

On the other hand, the Heliscope and PacBio SMRT systems do not need any pre-amplification steps as these systems are sensitive enough to detect individual single molecule template extensions [30]. The different strategies to generate the sequence reads also lead to differences in the output capacity for the different platforms. The performance comparison between high-throughput sequencing platforms is shown in Table 1. NGS technologies, also known as massively parallel sequencing or deep sequencing [31], include the second generation and third generation sequencing technology. At present, representative second-generation sequencing platforms are Roches 454 in Switzerland, genome analyzer (GA), HiSeq 2500 and MiSeq from Illumina in the United States, sequencing of oligomer connection detection in American ABI company (sequencing by oligo ligation detection, SOLiD) 5500xl, Ion Torrent personal genome machine (PGM) from Life Technologies, USA. The third generation sequencing platforms include the SMRT sequencing technology from Pacific Biosciences and the Nanopore sequencing technology from Oxford Nanopore Technologies, UK. Below we focus on the newer sequencing platforms, such as the Illumina, Life Technologies Semiconductor sequencing, PacBio and Nanopore.

The second generation sequencing technology

The second generation sequencing technologies are characterized by higher parallelism of operations, higher yield, simpler operation, much lower cost per read, and unfortunately shorter reads. Although the second generation sequencing platform produces highly accurate reads [32], the read may also lead to misassembly. The second generation sequencing platforms have characteristic error profiles that change as the technologies improve. Error profiles can include enrichment of base call error toward the 3 ends of reads, compositional bias for or against high-GC sequence, and inaccurate determination of simple sequence repeats [33]. The rates and types of sequencing errors with the next-generation platforms and library preparation method, *e.g.*, DNA sequence reads from Illumina sequencing technologies, have errors at the rate of 0.02%–0.05%. The second generation sequencing technology has the following characteristics compared with the first generation sequencing technology: (i) higher sequencing throughput. It does not rely on traditional capillary electrophoresis, and its sequencing reaction is performed on a chip, enabling simultaneous sequencing of millions of dots on the chip [34]; (ii) lower sequencing costs. It reduces the base cost per Mb by 96% to 99% compared to the Sanger sequencing method [35]; (iii) higher sensitivity. It has high sensitivity to identify signals with lower abundance; (iv) it is not convenient to follow up data analysis [36]; (v) more bias and mismatches. PCR process may introduce bias and mismatch [37].

Roche 454 sequencing platform

The Roche 454 system was the first next-generation sequencing platform available as a commercial product [38]. The Roche 454 system is performed by the

Table 1 The performance comparison between high-throughput sequencing platforms

Platform	Company	Error rate (%)	Read length (bp)	No. of reads/run	Time/run	Cost/Gb
GS FLX	454 Life Sciences, Roche	1	200–1000	0.4–0.5 Gb	~23 h	$9.5
SOLiD 5500xl	Applied Biosystems	0.1	2×35–2×75	30–50 Gb	~10 d	$70
Illumina HiSeq 2500	Solexa, Illumina	0.2	2×50–2×150	750–1500 Gb	~40 h	$45
Illumina MiSeq	Solexa, Illumina	0.2	2×50–2×300	7.5–13 Gb	21–56 h	$110
PacBio RS	Pacific biosciences	16	~20 $\times 10^3$	500 Mb–1 Gb	~4 h	$1000
Nanopore MinION	Oxford Nanopore	38	~200 $\times$ 10^3	500 Mb–1.5 Gb	~50 h	$750

No. of reads/run: the number of reads is generated by per run; Time/run: the time spent per run; Cost/Gb: the dollars spent per Gb.

pyrosequencing method [39]. Compared with other second generation sequencing platform, the Roche 454 system has the advantage of longer read length. The read generated by the Roche 454 system can be up to 1 kb in length. The accuracy of Roche 454 system was up to 99.9%, which reached the same accuracy as Sanger sequencing. Although the cost of sequencing for the 454 platform is much higher than other second generation sequencing platforms, it is still the most ideal choice for applications that requires long read length, such as *de novo* assembly.

Illumina sequencing platform

Illumina is currently the leader in the NGS industry and most library preparation protocols are compatible with the Illumina system. In addition, Illumina offers the highest throughput of all platforms and the lowest per-base cost [40]. Its length is up to 300 bp, compatible with almost all types of applications. The Illumina platform uses bridge amplification for polony generation and sequencing by a synthesis approach. Forward and reverse oligos for amplification, complementary to the adapter sequences introduced during the library preparation steps, are attached to the entire inside surface of the flow cell lanes. The bridge amplification scheme that Illumina exploits yields a high number of clusters, *i.e.*, with good loading of the flow cell, the total number of reads generated per HiSeq2000 lane may reach ~180 million. With a paired-end 2 × 100 bp read format the total output of one flow-cell lane is up to ~36 Gb. A full run of 2 flow cells sequencing in parallel may yield ~600 Gb of data [41].

SOLiD sequencing platform

The SOLiD system is widely claimed to have low error rates, 99.94% accuracy while most other systems are owing to the fact that each base is read twice [42]. SOLiD is the highest throughput system on the market. The disadvantage of SOLiD is that its reads are the shortest among all the platforms (75 bp maximum) and it takes relatively long run times [43]. The deficiencies of SOLiD makes it not well meet the needs of *de novo* assembly [44]. The SOLiD system is much less widely used than the Illumina system and the panel of sample preparation kits and services is less well developed.

The third generation sequencing technology

An important advantage of the third generation sequencing is the read length. While the original PacBio RS system with the first generation of chemistry generated mean read lengths around 1500 bp [45], the PacBio RS II system with C4 chemistry boasts average read lengths over 10 kb, with an N50 of more than 20 kb and maximum read lengths over 60 kb [46]. In contrast, the maximum read length of Illumina HiSeq2500 is only paired-end 250 bp [47]. The short read lengths of the second generation sequencing are commonly unable to span repetitive regions with at least one unique flanking sequence. In these cases, the origin of a read cannot be precisely determined. The highly-contiguous *de novo* assembly using PacBio sequencing can close gaps in current reference assemblies and characterize structural variation (SV) in personal genomes. With longer reads, we can sequence through extended repetitive regions and detect mutations, many of which are associated with diseases [48]. The limited yield, high error rate and high cost per base currently prohibit large scale sequencing projects on the third generation sequencing technologies. Currently, there are two main types of long read sequencing technologies: single-molecule realtime sequencing approaches and synthetic approaches that rely on existing short read technologies to construct long reads *in silico* [49].

PacBio sequencing platform

PacBio RS is a single-molecule real-time (SMRT) sequencing system developed by Pacific BioSciences. PacBio RS sequencing platform is developed by Pacific BioSciences, which offers longer reads (20 kb and even longer) than the second-generation sequencing technologies, making it wellsuited for unsolved problems in genome, transcriptome, and epigenetics research [50]. Another advantage is that its run times is short (30× human genome is expected to be completed in one day). The disadvantages of PacBio RS are higher costs (US$2–17 per Mb), higher error rates (~14%) and the lowest throughput among all platforms (maximum 500 Mb–1.5 Gb) [51]. All of these disadvantages have greatly limited the scop of its applications.

Oxford Nanopore sequencing platform

The most recent third-generation technology was released by Oxford Nanopore Technologies in 2014. Their current instrument, the Oxford Nanopore MinION is a handhold device that sequences DNA by electronically measuring the minute disruptions to electric current as DNA molecules pass through a nanopore. Nanopore sequencing is expected to offer solutions to the limitations of short read sequencing technologies and enable sequencing of large DNA molecules in minutes without having to modify or prepare samples [52]. Despite its potential many technical hurdles remain, Nanopore MinION is a small (~3 cm×10 cm) USBbased device that runs off a

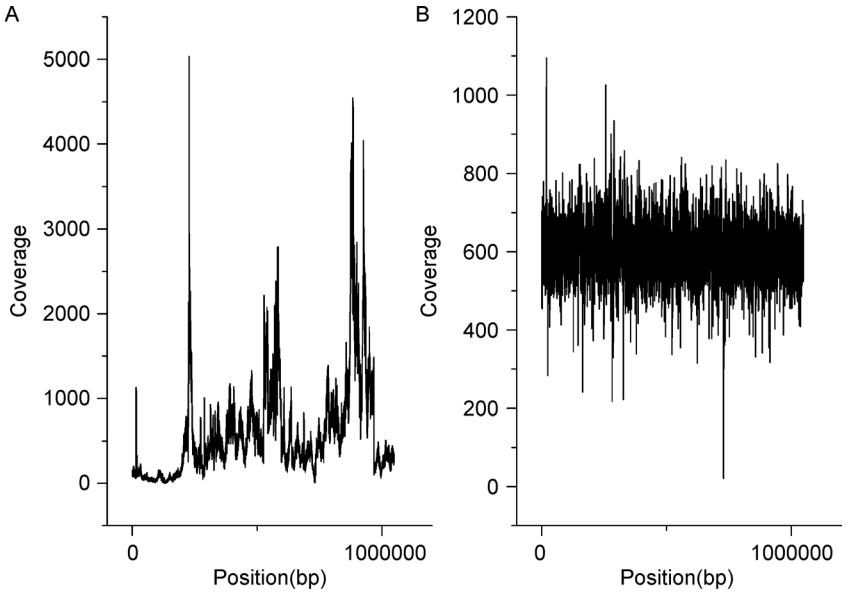

Figure 1. Coverage per genome positions in the *E. coli* datasets for lane1 (A), normal (B). The *y*-axis shows the number of reads that contain position *x* of the genome in res with the genome binned into 1000 bp windows [56]. The single-cell data sets of *E. coli* lane1 (subgraph A) (reads available at http://bix.ucsd.edu/singlecell/) display highly nonuniform coverage typical of single-cell amplification. In multi cell data set *E.coli* normal (subgraph B), most positions in the genome have coverage of 450–800×.

personal computer, giving it the smallest footprint of any current sequencing platform [53]. This affords the Nanopore MinION superior portability, highlighting its utility for rapid clinical responses and hard-to-reach field locations.

The single-cell sequencing technology

Cell theory provided an entirely new framework for understanding biology and diseases by asserting that cells are the basic unit of life [54]. Single-cell genomics aims to provide new perspectives to our understanding of genetics by bringing the study of genomes to the cellular level. Acquiring high-quality single-cell sequencing data has four primary technical challenges: efficient physical isolation of individual cells; amplification of the genome of a single cell to acquire sufficient material for downstream analysis; querying the genome in a cost-effective manner to identify variation that can test the hypotheses of the study; and interpreting the data within the context of biases and errors that are introduced during the first three steps. To maximize the quality of single-cell data and ensure that the signal is separable from technical noise, each of these variables requires careful consideration when designing single cell studies [55]. The single cell data sets display highly nonuniform coverage typical of single-cell amplification, including blackout regions, which are contiguous regions of the genome to which no reads aligned (coverage 0), just as shown in Figure 1, where the single-cell data sets of *E. coli* lane1 (subgraph A) (reads available at http://bix.ucsd.edu/singlecell/) display highly nonuniform coverage [56]. In multi cell data set *E. coli* normal (subgraph B), most positions in the genome have coverage of 450–800×.

DE NOVO ASSEMBLY METHODS

An assembly is a hierarchical data structure that maps the sequence data to a putative reconstruction of the target. It groups reads into contigs and contigs to scaffolds (sometimes called supercontigs or metacontigs, defining the contig order and orientation and the sizes of the gaps between contigs). According to existing literature, the assembly procedure can be classified as reference guided genome assembly and *de novo* genome assembly. *De novo* genome sequence assembly is important both to generate new sequence assemblies for previously uncharacterized genomes and identify the genome sequence of individuals in a reference-unbiased way. Here we focus on the comparison and evaluation of tools for *de novo* assembly of genome sequence. The next-generation assembly algorithms play around three basic frameworks for efficiently completing their tasks, namely, OLC graph [57], *de Bruijn* graph [58] and string graph [23]. The popular assemblers based on these three methods are summarized in Table 2. The illustration of the pipeline of *de novo* assembly is shown in Figure 2.

Overlap graph-based methods

OLC assembly algorithm generally works in three steps: first overlaps (O) among all the reads are found, then it carries out a layout (L) of all the reads and overlaps information on a graph and finally the consensus (C) sequence is inferred. It is an intuitionistic assembly algorithm, initially developed by Staden (1980) and subsequently extended and elaborated upon by many scientists [59]. Construction of OLC graph using example data from 15 bp length genome region is shown in Figure 2. During OLC assembly, overlaps between all reads are first detected, then contigs are formed by iteratively merging overlapping reads until a read heuristically determined to be at the boundary of a repeat is reached [60]. Repeats shorter than the minimally expected read overlap are often resolved, implying that genome resolution increases with read length. To account for sequencing errors, imprecise read overlaps are allowed, although this procedure may fragment the assembly even when the genomic repeats are nearly identical. The human genome was constructed primarily using OLC algorithms, and notable OLC-based assembly methods include parallel contig assembly program PCAP [61], AMOS [62], Arachne [27] and Celera [26].

De Bruijn graph-based methods

In *de Bruijn* graph, each node represents a *k*-mer (*k* consecutive bases in one read), and there will be a directed arc between two nodes if there is an overlap with *k*–1 bases and continuously emerge in one read. A read with the length of *r* can be divided into $r–k+1$ overlapping *k*-mers [63]. Construction of *de Bruijn* graph using example data from 15 bp length genome region is shown in Figure 2. The *de Bruijn* graph is classified into two types, Hamiltonian and Eulerian *de Bruijn* graphs, according to the method of expressing the nodes and edges [64]. In Hamiltonian approach, the *k*-mers are the nodes, whereas they are the edges in the Eulerian approach. The Hamiltonian graph approach is similar to the OLC approach in that the node is the sequence and the edge is the overlap. In Hamiltonian graph approach and OLC approach, the sequences are assembled by finding Hamiltonian paths that traverse all nodes, each of which is visited only once. This scenario is known as the NP-complete problem when the number of nodes is not trivial [65]. Normally, the computational complexity of finding the Hamiltonian paths is $O(m\times 2^n)$, where *m* is the total number of nodes, and *n* is the number of branching nodes [66]. The Hamiltonian approach is

Table 2 Summary of popular assemblers

Basic framework	Assembler	Input	Speed	Memory	N50
OLC graph	PCAP	SE/PE/Li/L	+	+	+++
	AMOS	SE/PE/Li	+	+	+++
	Arachne	SE/PE/Li	+	+	+++
	Celera	SE/PE/Li/L	+	+	+++
de Bruijn graph	Velvet	SE/PE/Li	++	++	+
	ALLPATHS	SE/PE/Li	+	+	+++
	Abyss	SE/PE/Li	++	+++	++
	SOAPdenovo2	SE/PE/Li	+++	++	++
	SparaseAssembler	SE/PE/Li	++	+++	++
	JR-Assembler	SE/PE/Li	+	+	+++
	MaSuRCA	SE/PE/Li/L	+	+	+++
	EPGA	PE	+	+++	++
	EPGA2	PE	+	+	++
	SPAdes	SE/PE/Li	++	+++	+++
	IDBA-UD	SE/PE/Li	++	+++	+++
	Velevet-SC	SE/PE/Li	++	+++	+++
	ALLPATHS-LG	PE/Li/L	+	+	+++
String graph	SGA	SE/PE/Li	+	+	+++
	Readjoiner	SE/PE/Li	+	+	+++
	FALCON	L	+	+++	++++

SE: single-end reads; PE: paired-end reads; Li: large-insert reads; L: long reads.

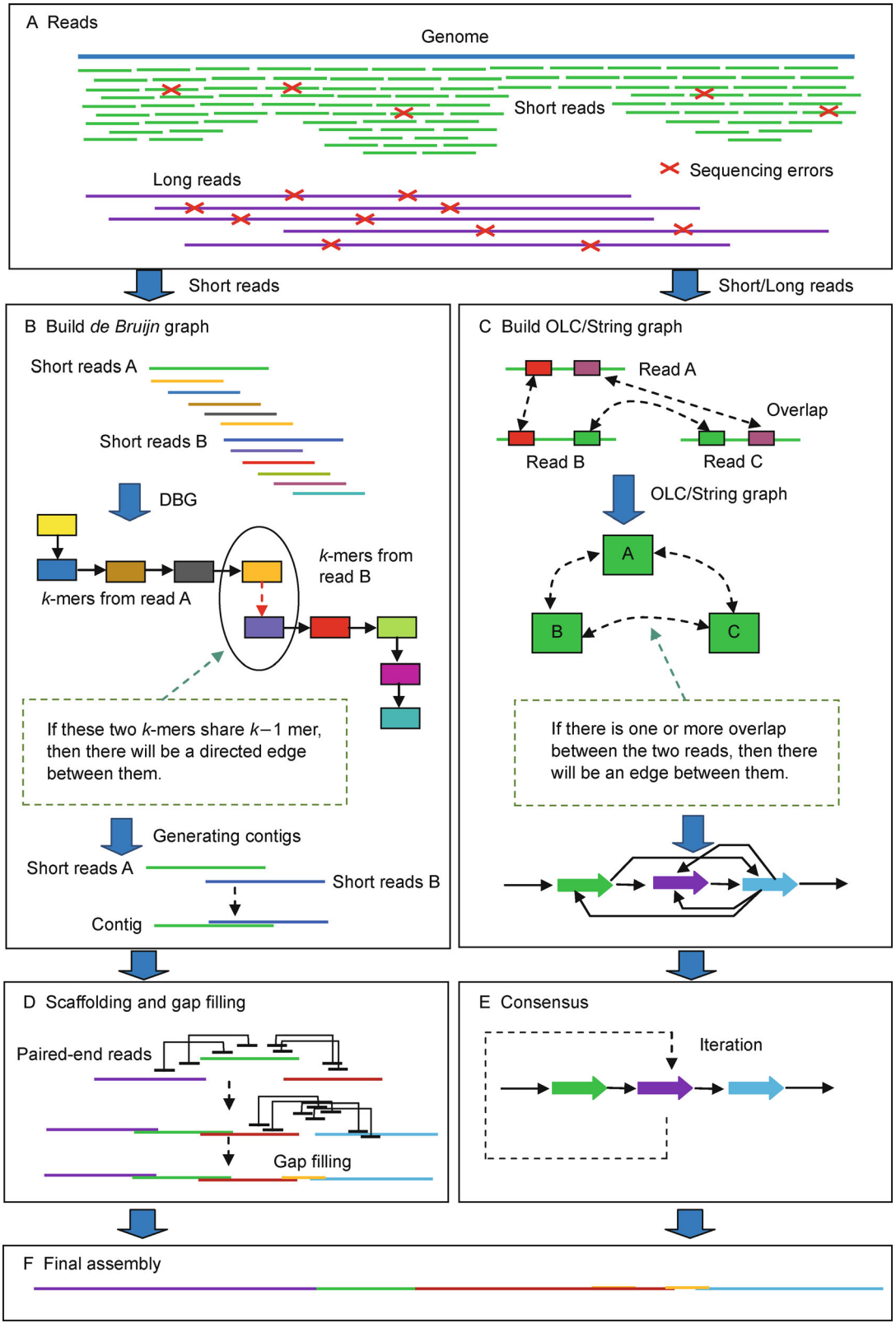

Figure 2. The illustration of the pipeline of *de novo* assembly. The subgraph (A) shows all reads; the subgraph (B) shows the principle of building *de Bruijn* graphs; The subgraph (C) shows the principle of building OLC/String graph; The subgraph (D) shows the principle of scaffolding and gap filling; The subgraph (E) shows the consensus operation; The subgraph (F) shows the final genome sequence.

widely used in *de novo* assembler such as SOAPdenovo [21], Abyss [18] and velvet [67]. In Eulerian graph approach, the sequences are assembled by finding Eulerian paths that traverse all edges, each of which is visited only once without simplification in polynomial time $O(n\times 2)$. The Eulerian approach is widely used in *de novo* assembler such as SPAdes [68], IDBA-UD [69], EPGA2 [70], MaSuRCA [71] and ALLPATHS [72]. The Eulerian *de Bruijn* graph based assemblers generally perform better in the assembly of a large genome than the Hamiltonian graph based assemblers. Construction of Hamiltonian and Eulerian *de Bruijn* graphs using example data from 6 bp length genome region are shown in Figure 3.

Assembly methods based on *de Bruijn* graphs begin, somewhat counter-intuitively, by replacing each read with the set of all-overlapping sequences of a shorter, fixed length [60]. The length is often denoted by *k*, and the sequences *k*-mers. The value of *k* is important for constructing *de Bruijn* graph. A large value of *k* will remove some short repetitive regions while reducing the number of nodes in *de Bruijn* graph, but will give rise to more unconnected sub-graphs which means that the number of gap regions increases. A small value of *k* will reduce some gap regions while increases the connectivity of *de Bruijn* graph, but will add more nodes and increase short repetitive regions. Therefore, the value of *k* cannot be too large or too small [70]. Contigs are formed by merging *k*-mers appearing adjacently in reads halting at *k*-mers from repeat boundaries. This has the cost of requiring highly accurate reads, and it initially discards some of the ability for reads to resolve repeats longer than *k* bases. It has the benefit of not requiring the storage of pairwise overlaps and having a graph structure representative of the repeat structure of the genome. For these reasons, *de Bruijn* graph is widely used in sequence assembly tools.

String graph-based methods

The string graph is the main data representation used by assemblers based on the OLC paradigm. Indeed, in a string graph, the vertices are the input reads and the arcs correspond to the overlapping reads, with the property that contigs are paths of the string graph. The string graph can be derived from the overlap graph by first removing duplicate reads (distinct elements of reads set with the same or reverse-complemented sequence) and contained reads (elements in reads set that are a substring of some element in reads set or their reverse complements), then removing transitive edges from the graph.

The string graph assembly formulation is similar in concept to a *de Bruijn* graph, however, it has the advantage of not decomposing sequences into *k*-mers but rather taking the full-length of a sequence read. The overlap-based approaches are more suitable than the *de Bruijn* graph-based methods for long sequences and single molecule sequencing reads of high error rate. They are produced based on operations of read overlap and the removal of transitively inferred overlaps. The most widely used string graph-based assembler is SGA [20], which fist constructs the Burrows and Wheeler Transform (BWT) [73] and the FM-index [74] of a set of reads, and then uses those data structures to efficiently compute the arcs of the string graph. Another famous string graph-based assembler is called FALCON [60] which was produced by Pacific Biosciences. FALCON is an experimental string graph-based assembler designed to preserve ambiguity in the assembly graph, and outputs the longest path through the graph along with alternate paths [75]. The disadvantage of FALCON is that it can be used only with high accuracy corrected sequences [76].

CHALLENGES AND SOLUTIONS

De novo genome assembly is an important issue in bioinformatics. With the advancement of next generation sequencing technologies, genome assembly has drawn more and more attention. Although a lot of genome assemblers are presented, there still exist four major challenges for *de novo* genome assembly using next generation reads. The first challenge is the sequencing errors, which possibly introduce artifacts in the assembly results. Sequencing errors usually lead to a complex *de Bruijn* graph. In general, the final results from a complex *de Bruijn* graph are often unsatisfactory; the second challenge is the sequencing bias. For example, Illumina sequencing platform has base composition bias (favoring

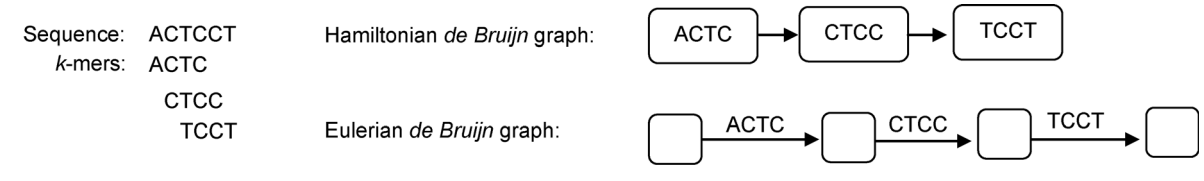

Figure 3. Construction of Hamiltonian and Eulerian *de Bruijn* graphs using example data from 6 bp length genome region. In this example, the length of genome sequence is 6 bp, the length of *k*-mer is 4 bp.

GC balanced regions), which usually results in uneven sequencing depth across genome; the third challenge is the topological complexity of repetitive regions in the genome. Most genomes contain a certain proportion of repeats, particulary mammalian in which repeats account for 25%–50% of its entire genome. The repeats cause not only misarrangements or gaps in the assembly results, but also the uneven depth of sequencing data. The last challenge is the huge computational resource consumption. Although the *de novo* assembly of small genomes, such as bacterial genomes, takes only serval minutes, the assembly of large genomes, such as mammalian genomes, typically takes several days to weeks and requires over tens to hundreds of GB of peak RAM memory.

Sequencing error

Rapid advances in next generation sequencing technology have led to exponential increase in the amount of genomic information. However, next generation sequencing reads contain far more errors than data from traditional sequencing methods, and downstream genomic analysis results can be improved by correcting the errors. The rates and types of sequencing errors vary according to the next generation sequencing platforms and library preparation methods. Below, we focus on the sequencing errors which are generated by the second and third generation sequencing platforms. Although the second generation sequencing platforms produce highly accurate reads (for example, sequence reads from Illumina sequencing technologies have errors at the rate of 0.5%–2.5%, and the errors tend to be accumulated in the 3 part of reads), the erroneous reads will not only result in misassemblies, but also lead to a complex *de Bruijn* graph. In general, the final results from a complex *de Bruijn* graph are often not satisfactory. There are three types of errors which widely reside in the second generation sequencing reads: substitutions, insertions and deletions. The description of those three types of error is shown in Figure 4.

In recent years, the third generation sequencing platforms have been widely used in various fields of genetic research. For the third generation sequencing platforms, the major error type is indel (insertions and deletions). For the third generation sequencing platforms, the average error rate is 15%, in some cases the error rate is as high as 30%. The comparison of error rate between different high throughput sequencing platforms is shown in Table 1.

The sequencing errors occur more frequently in regions with an extremely high GC or AT content, such as constant heterochromatin regions, including centromeres, telomeres or highly repetitive sequences, all of which may generate a complex assembly graph. Therefore, the sequencing errors should be corrected for more accurate and contiguous *de novo* assembly before or during assembly [77]. The error correction methods can be divided into three categories. The first one is *k*-mer counting-based method, the second one is multiple sequence alignment-based method, and the third one is probabilistic consistency-based method. Below, we introduce four methods one after one.

k-mer counting-based error correction

Most of the sequencing error correction tools implement the *k*-mer counting methods, and even tools in other categories often use *k*-mer counting to detect sequencing errors. If a genome is amplified through an ideal amplification processes, *k*-mers can be evenly distributed over all genome regions, and the histogram of *k*-mers depth forms a common distribution, *e.g.*, a Poisson (if the sequencing coverage is low) or Gaussian (if the sequencing coverage is high) distribution [78]. However, when sequencing errors and bias occur, the corresponding histogram of *k*-mers depth may have an exponentially decreasing or increasing curve, just as shown in Figure 5, where the left side shows the low depth *k*-mers which have a high probability of errors.

The *k*-mer counting-based error correction methods work by decomposing the reads into the set of all *k*-mers present in them, termed the *k*-specturm. In NGS data sets predominantly containing substitution errors, *k*-mers within a small Hamming distance from each other are likely to belong to the same genomic location. By identifying such a *k*-mer set, alignment is directly achieved without resorting to MSA, and error correction can then be applied by converting each consttituent *k*-mer to the consensus [5]. Typical *k*-mer counting-based error

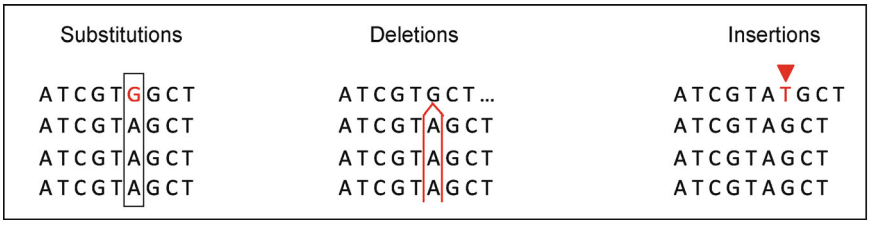

Figure 4. Three types of error widely reside in the second and third generation sequencing reads: substitutions, insertions and deletions.

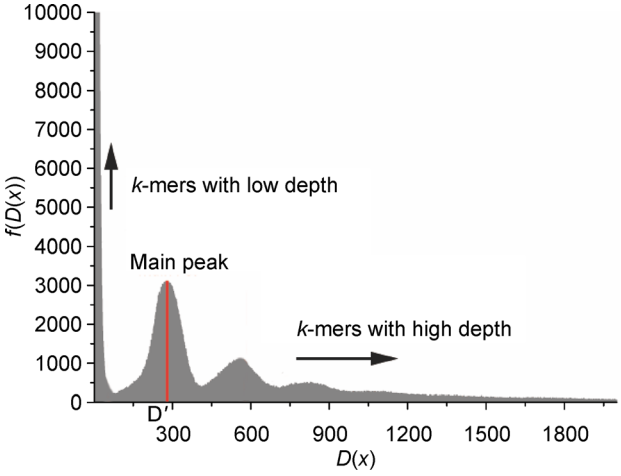

Figure 5. The *k*-mers histogram of staphylococcus aureus HiSeq (The dataset of staph is obtained from GAGE-B website, the size of *k*-mer is 11). The *x*-axis refers to the *k*-mer depth $D(x)$, which indicates "*k*-multiplet"; the *y*-axis refers to the frequency of the *k*-multiplet, $f(D(x))$. D' is the *k*-mer depth at the main peak of the *k*-mer histogram.

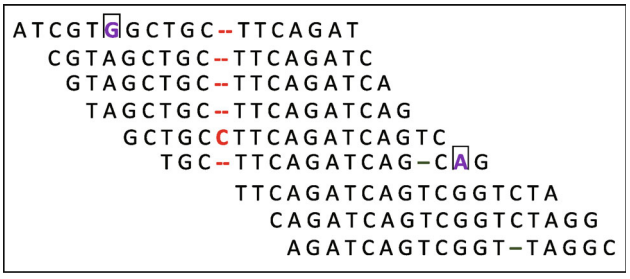

Figure 6. The principle of multiple sequence alignment-based error correction. The red letter in row 5 indicates insertion; the short green lines in rows 6 and 9 indicate deletions; the purple letters in rows 1 and 6 indicates substitutions. MSA based error correction methods work by aligning reads with each other and are corrected by consensus.

correction methods includes: Bless [77], Quack [79], Reptile [80], SOAPec2 [81], EDAR [82].

Multiple sequence alignment-based error correction

The multiple sequence alignment (MSA) based error correction methods work by aligning reads with each other and are corrected by consensus. Because those methods are based on MSA, they will consume more time to correct errors [83]. Typical MSA-based error correction methods for short reads include: RASCAL [84], Kalign [85], and Karect [86]: accurate correction of substitution, insertion and deletion errors for next-generation sequencing data. Although these methods are computationally expensive, they are commonly used for error correction of SMS long reads. Typical MSA based error correction methods for SMS long reads include: MECAT [76], LoRDEC [87]. The principle of MSA-based error correction is shown in Figure 6.

An excellent MSA should be able to quickly and accurately find the overlap of reads. To achieve this goal, researchers have paid a lot of efforts and achieved fruitful results. At present, all programs capable of genome wide SMS long reads alignment follow the seed-and-extend paradigm, seeding the alignment by using hash table index or more recently FM-index [74], and extending seed matches with the banded Smith-Waterman algorithm. This allows for sensitive detection of indels (insertions and deletions) as well as allowing for partial hits. Typical MSA-based algorithms for NGS short reads include: BWA [88], bowtie [89], MUMmer [90]. Typical MSA-based algorithms for SMS long reads include: SSAHA [91], MinHash [92], Min-Map [93], and MECAT [76].

Probabilistic consistency-based error correction

A common approach of error correction of reads is to determine a threshold and correct *k*-mers whose multiplicities fall below the threshold. Choosing the correct threshold is crucial since a low threshold results in too many uncorrected errors, while a high threshold results in the loss of correct *k*-mers. The histogram of the multiplicities of *k*-mers shows a mixture of two distributions) that of the errorfree *k*-mers, and that of the erroneous *k*-mers. When the coverage is high and uniform, these distributions are centered far apart and can be separated without much loss using a cutoff threshold; such methods therefore achieve excellent results [79]. Though high-throughput sequencing platforms provide relatively uniform coverage in many standard sequencing experiments, in some of the more challenging applications, such as single-cell sequencing, the coverage remains drastically uneven, just as shown in Figure 1. In the cases of uneven sequencing, the two methods described above lose their effect.

In recent years, some researchers have proposed error correction methods under the condition of uneven sequencing. These error correction methods use some special graph structures and simple statistical models to implement sequence error correction. Among them, the typical graph structures include Hamming graph [94], while the typical statistical models include Hidden Markov model [95] and Bayes model [96]. Typical probabilistic consistency-based error correction methods include Hammer [94], ProbCons [95], BayesHammer [96] and ECHO [97].

In some cases, the sequencing errors are discounted in the step of building the hash table of *k*-mers before the assembly step. For example, in order to effectively reduce assembly errors, EPGA [63] filters *k*-mers with unusually low frequencies. On the other hand, some assemblers are excluded by back trimming during extensions in the assembly steps, just like SPAdes [68] and IDBA-UD [69].

Uneven sequencing depth

Nonuniform coverage poses a serious problem for existing *de novo* assembly algorithms. Of *de Bruijn*-based assemblers, for example, Velvet [67], SOAPdenovo [21] and Abyss [18] use an average coverage cutoff threshold for contigs to prune out low-coverage regions, which tend to include more errors, whereas EULER-SR [98] uses a *k*-mer coverage cutoff. This pruning step not only reduces the complexity of the underlying *de Bruijn* graph substantially and makes then algorithms practical, but also seriously affects the effective length and genome fraction of the final assembly. Therefore, uneven sequencing is a major challenge of the current sequence assembly.

Cell theory provides an entirely new framework for understanding biology and disease by asserting that cells are the basic unit of life [54]. Single-cell genomics aims to provide new perspectives to our understanding of genetics by bringing the study of genomes to the cellular level. The single cell data sets display highly nonuniform coverage typical of single-cell amplification, including blackout regions, which are contiguous regions of the genome to which no reads aligned (coverage 0), just as shown in Figure 1. Most existing genome assemblers usually have an assumption that sequencing depths are even. These assemblers fail to construct correct long contigs. In order to solve this problem effectively, the researchers put forward some new solutions. Typical *de novo* assemblers for single-cell sequencing data include Velvet-SC [56], SPAdes [68] and IDBA-UD [69]. The flowchart of *de novo* assembly of single-cell sequencing data is shown in Figure 7.

Multisized *de Bruijn* graphs

The choice of *k* affects the construction of the *de Bruijn* graph. Smaller values of *k* collapse more repeats together, making the graph more tangled. Larger values of *k* may fail to detect overlaps between reads, particularly in low coverage regions, making the graph more fragmented.

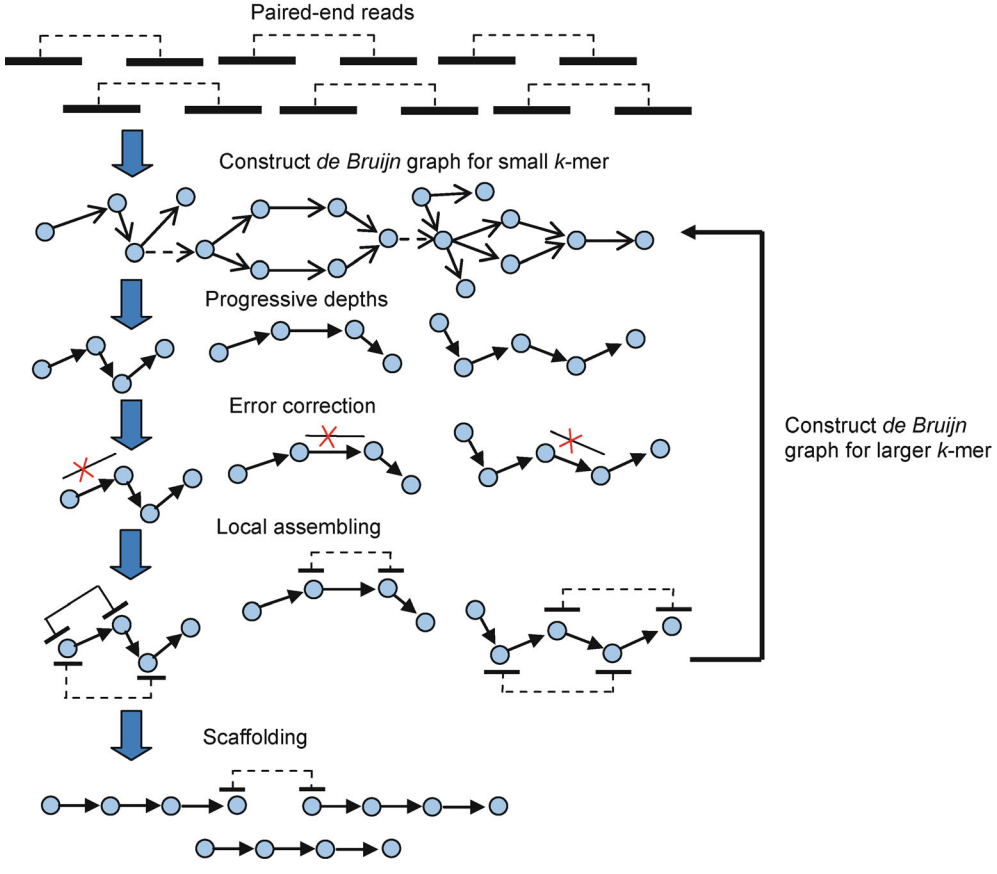

Figure 7. The flowchart of *de novo* assembly of single-cell sequencing data.

Since low coverage regions are typical for single cell sequencing data, the choice of k greatly affects the quality of single-cell assembly. Ideally, one should use smaller values of k in low coverage regions (to reduce fragmentation) and larger values of k in high coverage regions (to reduce repeat collapsing). The multisized *de Bruijn* graph allows us to vary k in this manner. The comparison of contigs which are generated by standard *de Bruijn* graph and multisized *de Bruijn* graph is shown in Figure 8.

Error correction in assembly graph

Errors in reads may lead to several types of structures in the *de Bruijn* graph. Miscalled bases and indels in the middle of a read typically lead to bulges. Bulges also arise from small variations between repeats in the genome. The bulge structures in the *de Bruijn* graph is shown in subgraph (A) in Figure 9. Errors near the ends of reads may lead to tips, just as shown in subgraph (B) in Figure 9. The chimeric reads may lead to erroneous connections in the graph, just as shown in subgraph (C) in Figure 9.

After obtaining the multisized *de Bruijn* graphs, it is usually possible to optimize the graph based on its topological structure. First, if nodes in one path have only one outgoing arc except the end node and only one ingoing arc except the start node, the path will be named simple path and can be merged into one node. Second, after merging simple paths, some tips (nodes whose out-degree plus in-degree is one) shorter than $2\times k$ can be removed. Tips are usually produced by erroneous bases in reads and gap regions. Third, there are some simple cycles (two nodes direct each other) and bubbles which can be simplified to one path. Bubbles or bulges are caused by non-exact repetitions in genomic sequences or biological variations, such as SNPs. On the graph, their structure is a redundant path, which diverges and then converges. Fixing a bubble involves removing the nodes that comprise the less-covered side, which simplifies the redundant paths into a single one.

Repetitive structures

Most genomes contain a certain proportion of repeats, particularly mammalian in which repeats account for 25%−50% of its entire genome [14]. Approximately 50% of the human genome comprises nonrandom repeat elements, such as LINEs, SINEs, LTRs and STRs [15],

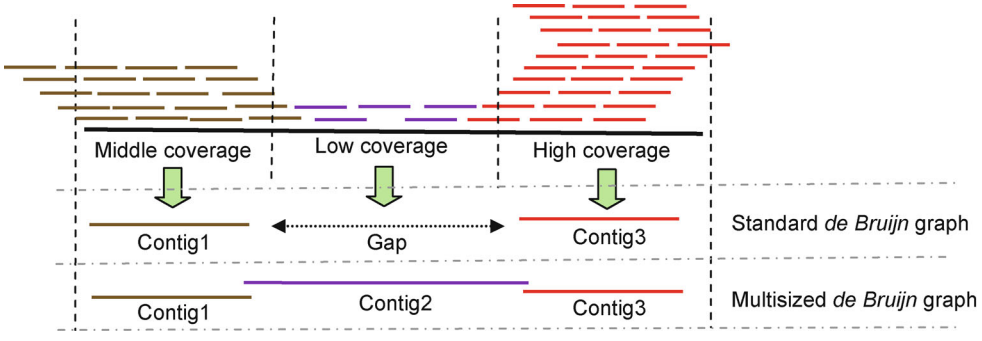

Figure 8. The comparison of contigs which are generated by standard *de Bruijn* graph and multisized *de Bruijn* graph.

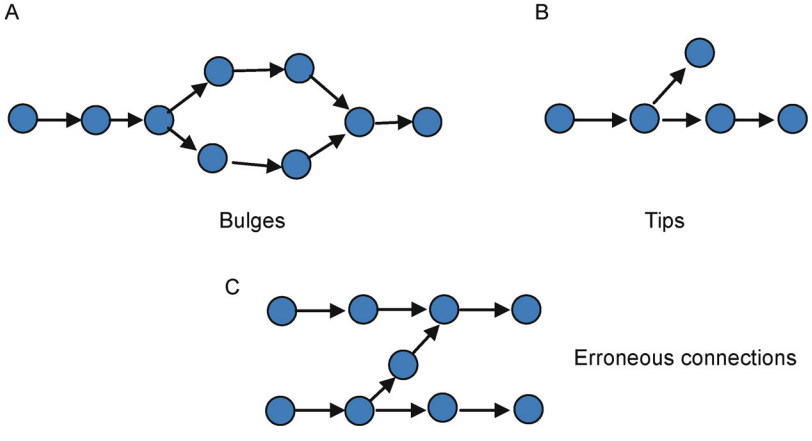

Figure 9. Errors in reads lead to several types of structures in the *de Bruijn* graph.

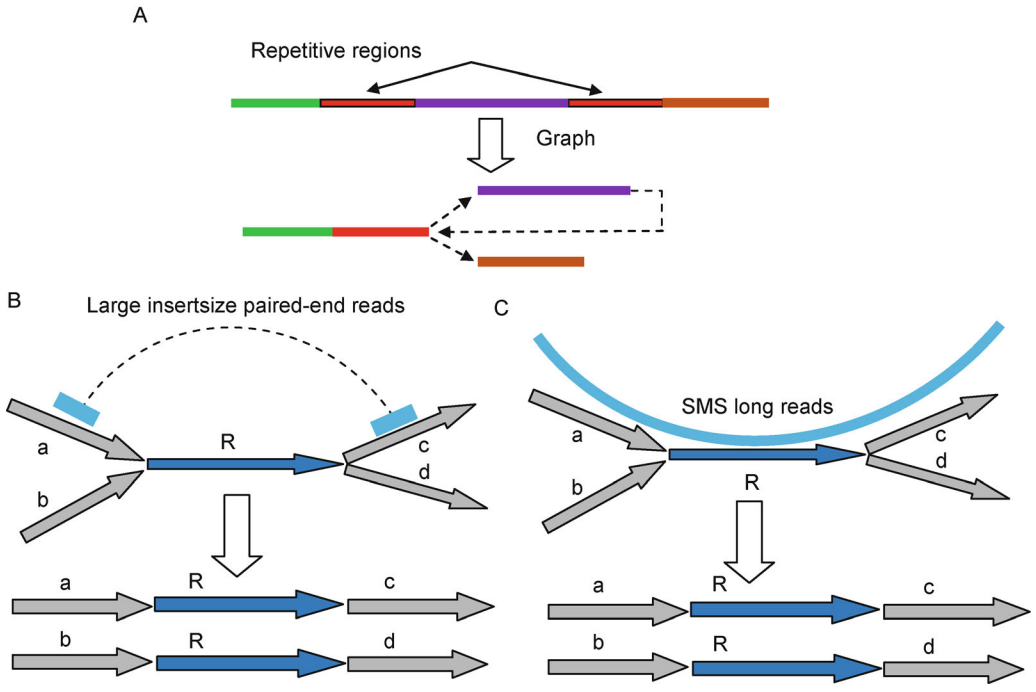

Figure 10. The structure of repetitive regions.

which often cause misarrangements or gaps in the assembly. The structure of repetitive regions is shown in Figure 10.

Today, there are two main methods to solve the problems caused by repetitive regions. The first one is to use the large insertsize paired-end reads; and the second one is to use the SMS long reads. Among then, the first method can only be used to solve the repetitive regions whose size is smaller than the insertsize [99–101]. The second method can only be used when the size of repetitive regions is smaller than the length of SMS long reads [102].

The size of insertsize is a few thousand bp [103], and the length of SMS long reads can reach tens of kb [104]. So the SMS long reads-based method can solve larger scale of repetitive regions than the large insertsize paired-end readsbased method. The large insertsize paired-end reads based method has been applied to almost all assembly tools. As the SMS long reads-based method requires the third generation sequencing data, it is mainly used in the assembly field of the combination of short and long reads. Although SMS long reads are of great help in solving the problem of repetitive regions, their range of efficacy is also limited. Faced with a huge size of repetitive regions (The length of SMS long reads is shorter than the length of repeat), the SMS long reads are also powerless.

The principle of the large insertsize paired-end reads based method is shown in subgraph (B) in Figure 10. The principle of the SMS long reads-based method is shown in subgraph (C) in Figure 10. The assemblers that use SMS long reads to solve the problems caused by repetitive regions includes SSPACE-Long Read [105], FinisherSC [106] and DBG2OLC [107]. In addition, repeats often cause misarrangements or gaps in the assembly, and its effects are difficult to eliminate under current technical conditions. In order to reduce the impact of repetitive regions on assembly as much as possible, researchers have proposed some new misassembly detection methods, such as MISSEQUEL [108], MEC [109] and PECC [110]. The principle of these methods is to improve the quality of draft genomes by identifying misassembly errors and their breakpoints with using paired-end sequence reads and optical mapping data.

Computational cost

Based on the Graph structure (such as *de Bruijn* graph or overlap graph), *de novo* assembly requires substantial RAM, storage and long computation times. For example, several short-read assembly packages(such as Abyss, ALLPATHS-LG [19], SGA and SOAPdenovo) have been proven for mammalian-size genomes up to the 3 Gbp human genome, and they generally take several days to weeks and require servers or clusters with 512 GB of RAM and many TB of disk space available for a gigabase-sized genome [22]. The specific resource consumption of these tools is shown in Table 3. Abyss

Table 3 The specific resources consumption of assemblers

Assembler	Resource consumption			
	Genome	Memory (GB)	Time (day)	Ref.
Abyss	Human	~16	~8	[18]
ALLPATHS-LG	Human	~512	~597	[19]
SGA	Human	~56	~1	[20]
SOAPdenovo2	Human	~35	~4	[21]

took 192 hours with Dell R720, 24 processers [18]. ALLPATHS-LG took 3.5 weeks with Dell R815, 48 processers [72]. SGA took 24 hours with single-Hexa-core XEON X5650 (2.66GHz) [20]. SOAPdenovo2 took 81 hours with eight Quad-cores AMD (2.3 GHz) [21]. Plant genomes are nearly 100 times larger than the currently sequenced bird, fish or mammalian genomes. In addition, they can have much higher ploidy, which is estimated to occur in up to 80% of all plant species and higher rates of heterozygosity and repeats their counterparts in other kingdoms. For all of these reasons, it will cost more resources for an assembler to assemble a plant genome than the mammalian genome. Advances in next generation sequencing technologies have resulted in the generation of unprecedented levels of sequence data, and traditional assembly tools have difficulty in processing large-scale data from high-throughput sequencing. In order to solve the new challenges caused by the dramatic increase in data volume, researchers have made some new explorations.

The message passing interface (MPI) and graphics processing unit (GPU) are pioneer programming application application programming interfaces (APIs) for parallel computing. Based on MPI technology, researchers have proposed some parallel assembly tools, such as Abyss [18] and Ray [111]. MPI cluster cannot deal with node failue. Given the absence of load balancing, fault tolerance and a distributed file system, MPI is unreliable and insufficiently robust.

The open source Apache Hadoop project, which adopts the MapReduce framework and a distributed file system, has recently given researchers an opportunity to achieve scalable, efficient and reliable computing performance on Linux clusters and on cloud computing services. MapReduce is an easy-to-use and general-purpose parallel programming model that is suitable for large scale data set analysis on a commodity hardware cluster developed by Google. The hadoop-based assembly tools include Contrail [112], CloudBrush [113], and DIME [114]. Contrail uses Hadoop for *de novo* assembly from short sequencing reads (without using a reference sequence), scaling up *de Bruijn* graph construction. CloudBrush is a *de novo* next generation genomic sequence assembler based on string graph and MapReduce cloud computing framework. DIME is a novel framework for *de novo* metegenomic sequence assembly based on Apache Hadoop platform. DIME offers great improvement in assembly across a range of sequence abundances and thus is robust to decreasing coverage.

OVERCOME THE CHALLENGES IN NGS ASSEMBLY BY USING SMS LONG READS

Generally, there are three important contributions of SMS long reads in NGS assembly. The first one is to provide guidance for resolving repetitive regions encountered during assembly (The first important contribution is mainly reflected in the contig expansion phase); the second one is to provide guidance for scaffolding (The second important contribution is mainly reflected in the scaffolding phase) and the last one is to provide guidance for gap filling (The third important contribution is mainly reflected in the gap filling phase). Below, we will give a detailed introduction to these three main contributions.

Provide guidance for solving repetitive structures

Although large-insert reads can mitigate the problems of repeats somewhat, they can not completely resolve the problems in genomic regions with long repeats (The span of paired-end reads is shorter than the length of repeat). Two new third generation single molecule sequencing technologies are currently available from Pacific Bioscience (PacBio) [115] and Oxford Nanopore [116]. The most established of these is the SMRT sequencing platform produced by PacBio. Their current instrument, the PacBio RS II, can generate reads as long as 54 kb with an average read length over 10 kb, approximately 50 to 250 times longer than those available from the widely used next generation Illumina platform [117]. In *de novo* assembly, the longer reads span more repetitive elements making it possible to assemble more contiguous sequences (contigs). Although SMS long reads are of great help in solving the problem of repetitive regions, their range of efficacy is also limited. Faced with a huge size of repetitive regions (The length of SMS long reads is shorter than the length of repeat), the SMS long reads are also powerless.

Provide guidance for scaffolding

The length of short read makes it difficult to obtain

complete *de novo* assemblies for genomes, which include longer repetitive sequences, so resolving the genomic order of some contiguous sequences (contigs) will be a challenge. However, the SMS long reads which produced by the third generation sequencing can be extended to tens of kilobases in length. It provides a new direction for overcoming this challenge. There are many scaffolders, which can combine contigs with each other, as guided by SMS long reads, such as SSPACE-LongRead [105], AHA [118], LINKS [119], OPERA-LG [120], DBG2OLC [107], hybridSPades [121]. The implementation details of these tools are different, but their core idea is to use the SMS long reads as the backbones to assist with scaffolding. The algorithms of these scaffolders have many advantages compared with the scaffolding by large-insert meta-pair reads. First of all, the short sequence read data and inability to scaffold across large repetitive structures translates into more gaps missing data and more incomplete reference assemblies [122]. However, the length of SMS long reads can be up to tens kilobases, they can easily cross the repetitive regions and effectively reduce the negative effect of the repetitive regions on the assembly results. Secondly, it easily solves the problems that occur during scaffolding with NGS reads with a larger inset size, because the SMS long reads tend to have less systematic and nucleotide composition biases and require less computational cost.

Provide guidance for gap filling

Sequencing biases, repetitive genomic features, genomic polymorphism, and other complicating factors all come together to make some regions difficult or impossible to assemble. Most of the gap closers are based on greedy-like extension processes and do not exhaustively search for the optimal solution; hence, the methods may fall into local minima. Some, but not all, gaps can be closed by existing gap closers, such as GapReduce [123], GapFiller [124], Sealer [125] or GapCloser [21]. Although existing gap filling tools can fill most of the gaps during assembly, their fill quality is not satisfactory. A recent study shows that the misassembly rates caused by the gap closer by using the NGS reads are 20–500 times higher than those by using SMS long read gap closer. Typical gap filling tools based on SMS long reads include: GMcloser [126], PBJelly [127]. It should be noted that strict error correction must be performed before gap filling with SMS long reads.

DISCUSSION

De novo genome assembly is an important issue in bioinformatics. With the advancement of next generation sequencing technologies, genome assembly has attracted more and more attention. Although a lot of genome assemblers are presented, there still exist several major challenges for *de novo* genome assembly by using NGS reads. The next generation sequencing technologies can be divided into three stages of development, which are the first generation, the second generation and the third generation. The first generation sequencing technology has gradually withdrawn from the stage of history, so we scarcely conduct too much research on it.

The second generation and the third generation sequencing technologies are currently widely used, so they become the focus of our research in this paper. The second generation sequencing technologies are characterized by higher parallelism of operations, higher yield, simpler operation, much lower cost per read, and unfortunately shorter reads. An important advantage of the third generation sequencing is the read length. While the original PacBio RS system with the first generation of chemistry generated mean read lengths around 1500 bp, the PacBio RS II system with C4 chemistry boasts average read lengths over 10 kb, with an N50 of more than 20 kb. The disadvantages of the third generation sequencing technologies are higher costs (US$2–17 per Mb), higher error rates (15%–30%) and the lowest throughput among all platforms (maximum 500 Mb–1.5 Gb). The limited yield and high cost per base currently prohibit large scale sequencing projects on the third generation sequencing technologies.

The differences in sequencing technologies have led to different assembly tools. For the second generation assembly tools, they assembled shorter segments, but with higher accuracy. Due to the short assembly segment, the second generation assemblers are difficult to solve the problem caused by repetitive regions. For the third generation assembly tools, they assemble longer segments, but with higher error rates. Due to the higher error rate of long reads, the third generation assemblers are difficult to achieve highprecision assembly. Although the third generation assemblers have the drawbacks of low accuracy, high cost and low throughput, they are highly helpful in resolving the problems as they generate more contiguous results.

Future sequencing technologies may also offer improvements. Currently, the original PacBio long read sequencing is expensive. However, with the advent of Oxford Nanopore MinION and PacBio sequel platforms, inexpensive long read sequencing technologies are within reach and may lead to long-read-only assembly being more frequently used. It has also been expected that quantum sequencing technologies may further reduce the cost problem by increasing the throughput and read length. The throughput of quantum sequencing should reach 100 Tb per day by 2018. Although this throughput is only an expectation, it appears promising that the

throughput problem of long SMS reads can be solved. Once the high throughput sequencing of long reads becomes a reality, the current long read assemblers would not be suitable, owing to overflowing memory, and thus a high-priority challenge in *de novo* assembly will be the development of new assembly algorithms with efficient memory and computational costs.

ACKNOWLEDGMENTS

This work has been supported by the National Natural Science Foundation of China (Nos. 61732009, 61772557 and 61420106009), supported by 111 Project (No. B18059) and the Fundamental Research Funds for the Central Universities of Central South University (No. 1053320171177).

COMPLIANCE WITH ETHICS GUIDELINES

The authors Xingyu Liao, Min Li, You Zou, Fang-Xiang Wu, Yi-Pan and Jianxin Wang declare that they have no conflict of interests.

This paper is a review and does not contain any studies with human or animal subjects performed by any of the authors.

REFERENCES

1. Miller, J. R., Koren, S. and Sutton, G. (2010) Assembly algorithms for next-generation sequencing data. Genomics, 95, 315–327
2. Nagarajan, N. and Pop, M. (2013) Sequence assembly demystified. Nat. Rev. Genet., 14, 157–167
3. Denton, J. F., Lugo-Martinez, J., Tucker, A. E., Schrider, D. R., Warren, W. C. and Hahn, M. W. (2014) Extensive error in the number of genes inferred from draft genome assemblies. PLoS Comput. Biol., 10, e1003998
4. Head, S. R., Komori, H. K., LaMere, S. A., Whisenant, T., Van Nieuwerburgh, F., Salomon, D. R. and Ordoukhanian, P. (2014) Library construction for next-generation sequencing: overviews and challenges. Biotechniques, 56, 61–64
5. Yang, X., Chockalingam, S. P. and Aluru, S. (2013) A survey of error-correction methods for next-generation sequencing. Brief. Bioinform., 14, 56–66
6. Kelley, D. R., Schatz, M. C. and Salzberg, S. L. (2010) Quake: quality-aware detection and correction of sequencing errors. Genome Biol., 11, R116
7. Koren, S. and Phillippy, A. M. (2015) One chromosome, one contig: complete microbial genomes from long-read sequencing and assembly. Curr. Opin. Microbiol., 23, 110–120
8. Madoui, M. A., Engelen, S., Cruaud, C., Belser, C., Bertrand, L., Alberti, A., Lemainque, A., Wincker, P. and Aury, J. M. (2015) Genome assembly using Nanopore-guided long and error-free DNA reads. BMC Genomics, 16, 327
9. Sims, D., Sudbery, I., Ilott, N. E., Heger, A. and Ponting, C. P. (2014) Sequencing depth and coverage: key considerations in genomic analyses. Nat. Rev. Genet., 15, 121–132
10. Chitsaz, H., Yee-Greenbaum, J. L., Tesler, G., Lombardo, M. J., Dupont, C. L., Badger, J. H., Novotny, M., Rusch, D. B., Fraser, L. J., Gormley, N. A., *et al.* (2011) Efficient *de novo* assembly of single-cell bacterial genomes from short-read data sets. Nat. Biotechnol., 29, 915–921
11. Rodrigue, S., Malmstrom, R. R., Berlin, A. M., Birren, B. W., Henn, M. R. and Chisholm, S. W. (2009) Whole genome amplification and *de novo* assembly of single bacterial cells. PLoS One, 4, e6864
12. Liao, X., Li, M., Zou, Y., Wu, F., Pan, Y., Luo, F., and Wang, J. (2018) Improving *de novo* assembly based on read classification. IEEE ACM T. Comput. Bi. http://dx.doi.org/10.1109/TCBB.2018.2861380
13. Margulies, M., Egholm, M., Altman, W. E., Attiya, S., Bader, J. S., Bemben, L. A., Berka, J., Braverman, M. S., Chen, Y. J., Chen, Z., *et al.* (2005) Genome sequencing in microfabricated high-density picolitre reactors. Nature, 437, 376–380
14. Kazazian, H. H. Jr. (2004) Mobile elements: drivers of genome evolution. Science, 303, 1626–1632
15. Cordaux, R. and Batzer, M. A. (2009) The impact of retrotransposons on human genome evolution. Nat. Rev. Genet., 10, 691–703
16. Goodwin, S., Gurtowski, J., Ethe-Sayers, S., Deshpande, P., Schatz, M. C. and McCombie, W. R. (2015) Oxford Nanopore sequencing, hybrid error correction, and *de novo* assembly of a eukaryotic genome. Genome Res., 25, 1750–1756
17. Oikonomopoulos, S., Wang, Y. C., Djambazian, H., Badescu, D. and Ragoussis, J. (2016) Benchmarking of the Oxford Nanopore MinION sequencing for quantitative and qualitative assessment of cDNA populations. Sci. Rep., 6, 31602
18. Simpson, J. T., Wong, K., Jackman, S. D., Schein, J. E., Jones, S. J. and Birol, I. (2009) ABySS: a parallel assembler for short read sequence data. Genome Res., 19, 1117–1123
19. Gnerre, S., Maccallum, I., Przybylski, D., Ribeiro, F. J., Burton, J. N., Walker, B. J., Sharpe, T., Hall, G., Shea, T. P., Sykes, S., *et al.* (2011) High-quality draft assemblies of mammalian genomes from massively parallel sequence data. Proc. Natl. Acad. Sci. USA, 108, 1513–1518
20. Simpson, J. T. and Durbin, R. (2012) Efficient *de novo* assembly of large genomes using compressed data structures. Genome Res., 22, 549–556
21. Luo, R., Liu, B., Xie, Y., Li, Z., Huang, W., Yuan, J., He, G., Chen, Y., Pan, Q., Liu, Y., *et al.* (2012) SOAPdenovo2: an empirically improved memory-efficient short-read *de novo* assembler. Gigascience, 1, 18
22. Schatz, M. C., Witkowski, J. and McCombie, W. R. (2012) Current challenges in *de novo* plant genome sequencing and assembly. Genome Biol., 13, 243
23. Idury, R. M. and Waterman, M. S. (1995) A new algorithm for DNA sequence assembly. J. Comput. Biol., 2, 291–306
24. Compeau, P. E. C., Pevzner, P. A. and Tesler, G. (2011) How to apply *de Bruijn* graphs to genome assembly. Nat. Biotechnol., 29, 987–991
25. Hernandez, D., François, P., Farinelli, L., Osterås, M. and Schrenzel, J. (2008) *de novo* bacterial genome sequencing:

millions of very short reads assembled on a desktop computer. Genome Res., 18, 802–809
26. Myers, E. W., Sutton, G. G., Delcher, A. L., Dew, I. M., Fasulo, D. P., Flanigan, M. J., Kravitz, S. A., Mobarry, C. M., Reinert, K. H., Remington, K. A., *et al.* (2000) A whole-genome assembly of *Drosophila*. Science, 287, 2196–2204
27. Jaffe, D. B., Butler, J., Gnerre, S., Mauceli, E., Lindblad-Toh, K., Mesirov, J. P., Zody, M. C. and Lander, E. S. (2003) Whole-genome sequence assembly for mammalian genomes: Arachne 2. Genome Res., 13, 91–96
28. Sohn, J. I. and Nam, J. W. (2018) The present and future of *de novo* whole-genome assembly. Brief. Bioinformatics, 19, 23–40
29. Mitra, R. D. and Church, G. M. (1999) *In situ* localized amplification and contact replication of many individual DNA molecules. Nucleic Acids Res., 27, e34–e39
30. Buermans, H. P. J. and den Dunnen, J. T. (2014) Next generation sequencing technology: advances and applications. Biochim. Biophys. Acta, 1842, 1932–1941
31. Metzker, M. L. (2010) Sequencing technologies–the next generation. Nat. Rev. Genet., 11, 31–46
32. Laehnemann, D., Borkhardt, A. and McHardy, A. C. (2016) Denoising DNA deep sequencing data-high-throughput sequencing errors and their correction. Brief. Bioinform., 17, 154–179
33. Schirmer, M., Ijaz, U. Z., D'Amore, R., Hall, N., Sloan, W. T. and Quince, C. (2015) Insight into biases and sequencing errors for amplicon sequencing with the Illumina MiSeq platform. Nucleic Acids Res., 43, e37–e37
34. van Dijk, E. L., Auger, H., Jaszczyszyn, Y. and Thermes, C. (2014) Ten years of next-generation sequencing technology. Trends Genet., 30, 418–426
35. Mestan, K. K., Ilkhanoff, L., Mouli, S. and Lin, S. (2011) Genomic sequencing in clinical trials. J. Transl. Med., 9, 222
36. Goodwin, S., McPherson, J. D. and McCombie, W. R. (2016) Coming of age: ten years of next-generation sequencing technologies. Nat. Rev. Genet., 17, 333–351
37. Quail, M. A., Smith, M., Coupland, P., Otto, T. D., Harris, S. R., Connor, T. R., Bertoni, A., Swerdlow, H. P. and Gu, Y. (2012) A tale of three next generation sequencing platforms: comparison of Ion Torrent, Pacific Biosciences and Illumina MiSeq sequencers. BMC Genomics, 13, 341
38. Schuster, S. C. (2008) Next-generation sequencing transforms today's biology. Nat. Methods, 5, 16–18
39. Patel, R. K. and Jain, M. (2012) NGS QC Toolkit: a toolkit for quality control of next generation sequencing data. PLoS One, 7, e30619
40. Liu, L., Li, Y., Li, S., Hu, N., He, Y., Pong, R., Lin, D., Lu, L. and Law, M. (2012) Comparison of next-generation sequencing systems. J. Biomed. Biotechnol., Article ID 251364
41. Liu, L., Hu, N., Wang, B., Min, C., Juan, W., Tian, Z., Yi, H. and Dan, L. (2011). A brief utilization report on the Illumina HiSeq 2000 sequencer. Mycology, 2, 169–191
42. Simon, S. A., Zhai, J., Nandety, R. S., McCormick, K. P., Zeng, J., Mejia, D. and Meyers, B. C. (2009) Short-read sequencing technologies for transcriptional analyses. Annu. Rev. Plant Biol., 60, 305–333
43. Kircher, M. and Kelso, J. (2010) High-throughput DNA sequencing–concepts and limitations. BioEssays, 32, 524–536
44. Hert, D. G., Fredlake, C. P. and Barron, A. E. (2008) Advantages and limitations of next-generation sequencing technologies: a comparison of electrophoresis and non-electrophoresis methods. Electrophoresis, 29, 4618–4626
45. Henson, J., Tischler, G. and Ning, Z. (2012) Next-generation sequencing and large genome assemblies. Pharmacogenomics, 13, 901–915
46. Rhoads, A. and Au, K. F. (2015) PacBio sequencing and its applications. Genomics Proteomics Bioinformatics, 13, 278–289
47. Logares, R., Haverkamp, T. H. A., Kumar, S., Lanzén, A., Nederbragt, A. J., Quince, C. and Kauserud, H. (2012) Environmental microbiology through the lens of high-throughput DNA sequencing: synopsis of current platforms and bioinformatics approaches. J. Microbiol. Methods, 91, 106–113
48. Treangen, T. J. and Salzberg, S. L. (2011) Repetitive DNA and next-generation sequencing: computational challenges and solutions. Nat. Rev. Genet., 13, 36–46
49. Heather, J. M. and Chain, B. (2016) The sequence of sequencers: The history of sequencing DNA. Genomics, 107, 1–8
50. Chin, C. S., Alexander, D. H., Marks, P., Klammer, A. A., Drake, J., Heiner, C., Clum, A., Copeland, A., Huddleston, J., Eichler, E. E., *et al.* (2013) Nonhybrid, finished microbial genome assemblies from long-read SMRT sequencing data. Nat. Methods, 10, 563–569
51. Ferrarini, M., Moretto, M., Ward, J. A., Šurbanovski, N., Stevanović, V., Giongo, L., Viola, R., Cavalieri, D., Velasco, R., Cestaro, A., *et al.* (2013) An evaluation of the PacBio RS platform for sequencing and *de novo* assembly of a chloroplast genome. BMC Genomics, 14, 670
52. Goodwin, S., Gurtowski, J., Ethe-Sayers, S., Deshpande, P., Schatz, M. C. and McCombie, W. R. (2015) Oxford Nanopore sequencing, hybrid error correction, and *de novo* assembly of a eukaryotic genome. Genome Res., 25, 1750–1756
53. Laver, T., Harrison, J., O'Neill, P. A., Moore, K., Farbos, A., Paszkiewicz, K. and Studholme, D. J. (2015) Assessing the performance of the Oxford Nanopore technologies minion. Biomol Detect. Quantif., 3, 1–8
54. Turner, W. (1890) The cell theory, past and present. J. Anat. Physiol., 24(Pt 2), 253–287
55. Gawad, C., Koh, W. and Quake, S. R. (2016) Single-cell genome sequencing: current state of the science. Nat. Rev. Genet., 17, 175–188
56. Chitsaz, H., Yee-Greenbaum, J. L., Tesler, G., Lombardo, M. J., Dupont, C. L., Badger, J. H., Novotny, M., Rusch, D. B., Fraser, L. J., Gormley, N. A., *et al.* (2011) Efficient *de novo* assembly of single-cell bacterial genomes from short-read data sets. Nat. Biotechnol., 29, 915–921
57. Batzoglou, S., Jaffe, D. B., Stanley, K., Butler, J., Gnerre, S., Mauceli, E., Berger, B., Mesirov, J. P. and Lander, E. S. (2002) ARACHNE: a whole-genome shotgun assembler. Genome Res.,

12, 177–189
58. Compeau, P. E. C., Pevzner, P. A. and Tesler, G. (2011) How to apply *de Bruijn* graphs to genome assembly. Nat. Biotechnol., 29, 987–991
59. Li, Z., Chen, Y., Mu, D., Yuan, J., Shi, Y., Zhang, H., Gan, J., Li, N., Hu, X., Liu, B., *et al.* (2012) Comparison of the two major classes of assembly algorithms: overlap-layout-consensus and *de-bruijn*-graph. Brief. Funct. Genomics, 11, 25–37
60. Chaisson, M. J. P., Wilson, R. K. and Eichler, E. E. (2015) Genetic variation and the *de novo* assembly of human genomes. Nat. Rev. Genet., 16, 627–640
61. Huang, X., Wang, J., Aluru, S., Yang, S. P. and Hillier, L. (2003) PCAP: a whole-genome assembly program. Genome Res., 13, 2164–2170
62. Treangen, T. J., Sommer, D. D., Angly, F. E., Koren, S. and Pop, M. (2011) Next generation sequence assembly with AMOS. Curr. Protoc. Bioinformatics. 33, 11.8. 1–11.8. 18
63. Luo, J., Wang, J., Zhang, Z., Wu, F. X., Li, M. and Pan, Y. (2015) EPGA: *de novo* assembly using the distributions of reads and insert size. Bioinformatics, 31, 825–833
64. Conway, T. C. and Bromage, A. J. (2011) Succinct data structures for assembling large genomes. Bioinformatics, 27, 479–486
65. Pevzner, P. (2000) Computational Molecular Biology: An Algorithmic Approach. Cambridge: MIT press
66. Pevzner, P. A., Tang, H. and Waterman, M. S. (2001) An Eulerian path approach to DNA fragment assembly. Proc. Natl. Acad. Sci. USA, 98, 9748–9753
67. Zerbino, D. R. and Birney, E. (2008) Velvet: algorithms for *de novo* short read assembly using *de Bruijn* graphs. Genome Res., 18, 821–829
68. Bankevich, A., Nurk, S., Antipov, D., Gurevich, A. A., Dvorkin, M., Kulikov, A. S., Lesin, V. M., Nikolenko, S. I., Pham, S., Prjibelski, A. D., *et al.* (2012) SPAdes: a new genome assembly algorithm and its applications to single-cell sequencing. J. Comput. Biol., 19, 455–477
69. Peng, Y., Leung, H. C. M., Yiu, S. M. and Chin, F. Y. (2012) IDBA-UD: a *de novo* assembler for single-cell and metagenomic sequencing data with highly uneven depth. Bioinformatics, 28, 1420–1428
70. Luo, J., Wang, J., Li, W., Zhang, Z., Wu, F. X., Li, M. and Pan, Y. (2015) EPGA2: memory-efficient *de novo* assembler. Bioinformatics, 31, 3988–3990
71. Zimin, A. V., Marçais, G., Puiu, D., Roberts, M., Salzberg, S. L. and Yorke, J. A. (2013) The MaSuRCA genome assembler. Bioinformatics, 29, 2669–2677
72. Butler, J., MacCallum, I., Kleber, M., Shlyakhter, I. A., Belmonte, M. K., Lander, E. S., Nusbaum, C. and Jaffe, D. B. (2008) ALLPATHS: *de novo* assembly of whole-genome shotgun microreads. Genome Res., 18, 810–820
73. Li, H. and Durbin, R. (2009) Fast and accurate short read alignment with Burrows-Wheeler transform. Bioinformatics, 25, 1754–1760
74. Simpson, J. T. and Durbin, R. (2010) Efficient construction of an assembly string graph using the FM-index. Bioinformatics, 26, i367–i373
75. Koren, S. and Phillippy, A. M. (2015) One chromosome, one contig: complete microbial genomes from long-read sequencing and assembly. Curr. Opin. Microbiol., 23, 110–120
76. Xiao, C. L., Chen, Y., Xie, S. Q., Chen, K-N, Wang, Y., Luo, F., and Xie, Z. (2016) MECAT: an ultra-fast mapping, error correction and *de novo* assembly tool for single-molecule sequencing reads. bioRxiv, 089250
77. Heo, Y., Wu, X. L., Chen, D., Ma, J. and Hwu, W. M. (2014) BLESS: bloom filter-based error correction solution for high-throughput sequencing reads. Bioinformatics, 30, 1354–1362
78. Li, X. and Waterman, M. S. (2003) Estimating the repeat structure and length of DNA sequences using L-tuples. Genome Res., 13, 1916–1922
79. Kelley, D. R., Schatz, M. C. and Salzberg, S. L. (2010) Quake: quality-aware detection and correction of sequencing errors. Genome Biol., 11, R116
80. Yang, X., Dorman, K. S. and Aluru, S. (2010) Reptile: representative tiling for short read error correction. Bioinformatics, 26, 2526–2533
81. Li, R., Zhu, H., Ruan, J., Qian, W., Fang, X., Shi, Z., Li, Y., Li, S., Shan, G., Kristiansen, K., *et al.* (2010) *De novo* assembly of human genomes with massively parallel short read sequencing. Genome Res., 20, 265–272
82. Zhao, X., Palmer, L. E., Bolanos, R., Mircean, C., Fasulo, D. and Wittenberg, G. M. (2010) EDAR: an efficient error detection and removal algorithm for next generation sequencing data. J. Comput. Biol., 17, 1549–1560
83. Salmela, L. and Schröder, J. (2011) Correcting errors in short reads by multiple alignments. Bioinformatics, 27, 1455–1461
84. Thompson, J. D., Thierry, J. C. and Poch, O. (2003) RASCAL: rapid scanning and correction of multiple sequence alignments. Bioinformatics, 19, 1155–1161
85. Lassmann, T. and Sonnhammer, E. L. L. (2005) Kalign–an accurate and fast multiple sequence alignment algorithm. BMC Bioinformatics, 6, 298
86. Allam, A., Kalnis, P. and Solovyev, V. (2015) Karect: accurate correction of substitution, insertion and deletion errors for next-generation sequencing data. Bioinformatics, 31, 3421–3428
87. Salmela, L. and Rivals, E. (2014) LoRDEC: accurate and efficient long read error correction. Bioinformatics, 30, 3506–3514
88. Li, H. and Durbin, R. (2009) Fast and accurate short read alignment with Burrows-Wheeler transform. Bioinformatics, 25, 1754–1760
89. Langmead, B., Trapnell, C., Pop, M. and Salzberg, S. L. (2009) Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. Genome Biol., 10, R25
90. Kurtz, S., Phillippy, A., Delcher, A. L., Smoot, M., Shumway, M., Antonescu, C. and Salzberg, S. L. (2004) Versatile and open software for comparing large genomes. Genome Biol., 5, R12
91. Ning, Z., Cox, A. J. and Mullikin, J. C. (2001) SSAHA: a fast search method for large DNA databases. Genome Res., 11, 1725–1729
92. Berlin, K., Koren, S., Chin, C. S., Drake, J. P., Landolin, J. M. and

Phillippy, A. M. (2015) Assembling large genomes with single-molecule sequencing and locality-sensitive hashing. Nat. Biotechnol., 33, 623–630
93. Li, H. (2016) Minimap and miniasm: fast mapping and *de novo* assembly for noisy long sequences. Bioinformatics, 32, 2103–2110
94. Medvedev, P., Scott, E., Kakaradov, B. and Pevzner, P. (2011) Error correction of high-throughput sequencing datasets with non-uniform coverage. Bioinformatics, 27, i137–i141
95. Do, C. B., Mahabhashyam, M. S. P., Brudno, M. and Batzoglou, S. (2005) ProbCons: probabilistic consistency-based multiple sequence alignment. Genome Res., 15, 330–340
96. Nikolenko, S. I., Korobeynikov, A. I. and Alekseyev, M. A. (2013) BayesHammer: Bayesian clustering for error correction in single-cell sequencing, BMC genomics. BioMed Central, 2013, S7
97. Kao, W. C., Chan, A. H. and Song, Y. S. (2011) ECHO: a reference-free short-read error correction algorithm. Genome Res., 21, 1181–1192
98. Chaisson, M. J. and Pevzner, P. A. (2008) Short read fragment assembly of bacterial genomes. Genome Res., 18, 324–330
99. Li, M., Liao, Z., He, Y., Wang, J., Luo, J. and Pan, Y. (2017) ISEA: iterative seed-extension algorithm for *de novo* assembly using paired-end information and insert size distribution. IEEE/ACM Trans. Comput. Biol. Bioinformatics, 14, 916–925
100. Luo, J., Wang, J., Zhang, Z., Li, M. and Wu, F. X. (2017) BOSS: a novel scaffolding algorithm based on an optimized scaffold graph. Bioinformatics, 33, 169–176
101. Li, M., Tang, L., Wu, F. X., Pan, Y. and Wang, J. (2018) SCOP: a novel scaffolding algorithm based on contig classification and optimization. Bioinformatics, doi: 10.1093/bioinformatics/bty773
102. Huddleston, J., Ranade, S., Malig, M., Antonacci, F., Chaisson, M., Hon, L., Sudmant, P. H., Graves, T. A., Alkan, C., Dennis, M. Y., *et al.* (2014) Reconstructing complex regions of genomes using long-read sequencing technology. Genome Res., 24, 688–696
103. Mostovoy, Y., Levy-Sakin, M., Lam, J., Lam, E. T., Hastie, A. R., Marks, P., Lee, J., Chu, C., Lin, C., Džakula, Ž., *et al.* (2016) A hybrid approach for *de novo* human genome sequence assembly and phasing. Nat. Methods, 13, 587–590
104. Chaisson, M. J. and Tesler, G. (2012) Mapping single molecule sequencing reads using basic local alignment with successive refinement (BLASR): application and theory. BMC Bioinformatics, 13, 238
105. Boetzer, M. and Pirovano, W. (2014) SSPACE-LongRead: scaffolding bacterial draft genomes using long read sequence information. BMC Bioinformatics, 15, 211
106. Lam, K. K., LaButti, K., Khalak, A. and Tse, D. (2015) FinisherSC: a repeat-aware tool for upgrading *de novo* assembly using long reads. Bioinformatics, 31, 3207–3209
107. Ye, C., Hill, C. M., Wu, S., Ruan, J. and Ma, Z. S. (2016) DBG2OLC: efficient assembly of large genomes using long erroneous reads of the third generation sequencing technologies. Sci. Rep., 6, 31900
108. Muggli, M. D., Puglisi, S. J., Ronen, R. and Boucher, C. (2015) Misassembly detection using paired-end sequence reads and optical mapping data. Bioinformatics, 31, i80–i88
109. Wu, B., Li, M., Liao, X., Luo, J., Wu, F., Pan, Y. and Wang, J. (2018) MEC: Misassembly Error Correction in contigs based on distribution of paired-end reads and statistics of GC-contents. IEEE/ACM Trans. Comput. Biol. Bioinformatics, 1
110. Li, M., Wu, B., Yan, X., Luo, J., Pan, Y., Wu, F. X. and Wang, J. (2017) PECC: Correcting contigs based on paired-end read distribution. Comput. Biol. Chem., 69, 178–184
111. Boisvert, S., Raymond, F., Godzaridis, E., Laviolette, F. and Corbeil, J. (2012) Ray Meta: scalable *de novo* metagenome assembly and profiling. Genome Biol., 13, R122
112. Schatz, M. C., Sommer, D., Kelley, D. and Pop, M. (2010) *De novo* assembly of large genomes using cloud computing. In Proceedings of the Cold Spring Harbor Biology of Genomes Conference
113. Chang, Y. J., Chen, C. C., Ho, J. M. and Chen, C. –L. (2012) *De novo* assembly of high-throughput sequencing data with cloud computing and new operations on string graphs. In Cloud Computing (CLOUD), 2012 IEEE 5th International Conference. pp. 155–161
114. Guo, X., Yu, N., Ding, X., Wang, J. and Pan, Y. (2015) DIME: a novel framework for *de novo* metagenomic sequence assembly. J. Comput. Biol., 22, 159–177
115. Roberts, R. J., Carneiro, M. O. and Schatz, M. C. (2013) The advantages of SMRT sequencing. Genome Biol., 14, 405
116. Sharma, T. R., Devanna, B. N., Kiran, K., Singh, P. K., Arora, K., Jain, P., Tiwari, I. M., Dubey, H., Saklani, B., Kumari, M., *et al.* (2018) Status and prospects of next generation sequencing technologies in crop plants. Curr. Issues Mol. Biol., 27, 1–36
117. Lee, H., Gurtowski, J., Yoo, S., Marcus, s., McCombie, W, and Schatz, M. (2014) Error correction and assembly complexity of single molecule sequencing reads. bioRxiv, 006395
118. Bashir, A., Klammer, A., Robins, W. P., Chin, C. S., Webster, D., Paxinos, E., Hsu, D., Ashby, M., Wang, S., Peluso, P., *et al.* (2012) A hybrid approach for the automated finishing of bacterial genomes. Nat. Biotechnol., 30, 701–707
119. Warren, R. L., Yang, C., Vandervalk, B. P., Behsaz, B., Lagman, A., Jones, S. J. and Birol, I. (2015) LINKS: scalable, alignment-free scaffolding of draft genomes with long reads. Gigascience, 4, 35
120. Gao, S., Bertrand, D., Chia, B. K. H. and Nagarajan, N. (2016) OPERA-LG: efficient and exact scaffolding of large, repeat-rich eukaryotic genomes with performance guarantees. Genome Biol., 17, 102
121. Antipov, D., Korobeynikov, A., McLean, J. S. and Pevzner, P. A. (2016) hybridSPAdes: an algorithm for hybrid assembly of short and long reads. Bioinformatics, 32, 1009–1015
122. Huddleston, J., Ranade, S., Malig, M., Antonacci, F., Chaisson, M., Hon, L., Sudmant, P. H., Graves, T. A., Alkan, C., Dennis, M. Y., *et al.* (2014) Reconstructing complex regions of genomes using long-read sequencing technology. Genome Res., 24, 688–

696

123. Luo, J., Wang, J., Shang, J., Luo, H., Li, M., Wu, F. and Pan, Y. (2018) GapReduce: a gap filling algorithm based on partitioned read sets. IEEE/ACM Trans. Comput. Biol. Bioinformatics, 1
124. Boetzer, M. and Pirovano, W. (2012) Toward almost closed genomes with GapFiller. Genome Biol., 13, R56
125. Paulino, D., Warren, R. L., Vandervalk, B. P., Raymond, A., Jackman, S. D. and Birol, I. (2015) Sealer: a scalable gap-closing application for finishing draft genomes. BMC Bioinformatics, 16, 230
126. Kosugi, S., Hirakawa, H. and Tabata, S. (2015) GMcloser: closing gaps in assemblies accurately with a likelihood-based selection of contig or long-read alignments. Bioinformatics, 31, 3733–3741
127. English, A. C., Richards, S., Han, Y., Wang, M., Vee, V., Qu, J., Qin, X., Muzny, D. M., Reid, J. G., Worley, K. C., *et al.* (2012) Mind the gap: upgrading genomes with Pacific Biosciences RS long-read sequencing technology. PLoS One, 7, e47768

