

Genome assembly and annotation

Pallavi Mishra^{1,2}, Ranjeet Maurya³, Himanshu Avashthi^{2,4}, Shikha Mittal⁴, Muktesh Chandra⁵ and Pramod Wasudeo Ramteke^{6,7}

¹Centre for Agriculture Bioinformatics, ICAR-Indian Agricultural Statistics Research Institute, India, ²Department of Computational Biology & Bioinformatics, Sam Higginbottom University of Agriculture Technology & Sciences, India, ³CSIR-Institute of Genomics and Integrative Biology, India, ⁴Division of Genomic Resources, ICAR National Bureau of Plant Genetic Resources, Pusa Campus, New Delhi, India, ⁵Centre of Bioinformatics, Institute of Interdisciplinary Sciences, University of Allahabad, Prayagraj, India, ⁶Department of Biological Sciences, Sam Higginbottom University of Agriculture Technology & Sciences, India, ⁷Faculty of Life Sciences, Mandsaur University, India

4.1 Introduction

Advancement in DNA sequencing technologies and cost reduction in sequencing have encouraged the development of genome assemblies for various organisms (Mardis, 2017). Genome assembly is the step where read fragments of a DNA sequence are aligned and combined to recreate the original DNA structure (Liu, Wu, LeVan, & SoftGenetics LLC, 2012). The longer sequence resulting from sequence assembly is called a “contig” sequence. Contigs are usually joined after the first assembly to construct long length of sequence, which is called as scaffolds (Ekblom & Wolf, 2014). Genome assembly is therefore includes a sequential process where it is carried out in steps starting with the assembly into contigs of the sequence reads and then the assembly into the scaffolds (supercontigs) followed by the assembly into chromosomes (Batzoglou et al., 2002). The gaps into the contigs filled with the unassigned base pairs with the character “N” that can be joined by gap-closing during genome finishing step (Catchen, Amores, & Bassham, 2020). Assembly is a crucial step in the process of genome analysis because with the present available technologies, the complete genome cannot be read with one step (Li et al., 2010). Assembly is a complex process since several separate pieces of the genome have to be stitched to put it together in the right order to make some informative sense (Baker, 2012). Often, due to duplication of the nucleotide bases in the genome, there is some possibility of error in the assembling that may give the incorrect assembly (Torresen et al., 2019). Two key approaches have been used to assemble genomes that include the de novo approach of assembly and the mapping or reference-based approach (Jin et al., 2020). To recreate full-length sequences without modifying the reference genome, it starts with the de novo assembly of short reads (Koren et al., 2012). The de novo assembly method constructs a genome by de novo assembling the individual fragment sequences and, to classify core gene sets, accompanied by whole genome comparisons (Ng & Kirkness, 2010). The reference-based assembly method works by mapping or aligning fragment sequences to reference genome and assembling those reads that do not mapped on the reference genome as the new contigs (Grabherr et al., 2011). Realignment of all the read sequences back to the reference genome enables to call for gene presence and absence. This approach needs lower sequencing depth than the de novo approach requires, which enables this a cost-effective method for large set of samples (Li et al., 2010). However, both the de novo and reference-based approaches of assembly give advantages in genome building but have limitations.

For example, the read depth is required comparatively high in the de novo method, which significantly increases the cost of sequencing (Koren et al., 2012). Assembly is then followed by an annotation step where the gene positions are localized and the functions are determined within the base sequences. Lately, more effective assembly tools for whole-genome analysis have been developed, facilitating even small labs to conduct whole-genome analysis for their genomics research (Schwarze, Buchanan, Taylor, & Wordsworth, 2018). At present the majority of genomic studies carried out on the basis of single reference genome comparison (Kurtz et al., 2004). Due to genetic differences between individuals, a reference genome constructed from one individual may not acquire all genomic information of that species (Wheeler et al., 2008). Genome assembly and annotation errors are more common. However, it necessitates a more complete reference genome covering all of the genomic information for particular species (Giang et al., 2020). Usually, these analysis

tools are diverse and use different algorithms, parameter settings, and input types; therefore it becomes difficult for researchers to choose and use it to get the exact result and, hence, these also cause errors in calling the presence and absence of genes (Bayer et al., 2017). In view of computational need, *de novo* assembly approach is high computationally intensive and requires substantial high-performance computing systems if many sample reads are assembling together (Su, Pan, Song, Xu, & Ning, 2014). Therefore genome assembly and annotation is difficult, particularly for complex and large genomes. With a couple of genome ventures, from microbes to eukaryotes, we are actually in the genome era. So, this chapter is emphasized on the approaches to genome assembly and annotation that construct a genome from its constituent reads and the computational techniques to annotate them to extract the necessary information from it.

4.1.1 How do you reassemble a genome after sequencing?

The DNA fragments that pass out of the sequencer after the completion of genome sequencing are all mixed up. So, we need to pick the fragments of the genome and bring them back together like a puzzle piece. To recreate the complete genome sequence and bring the fragments together in the right order, the alignment method is used called as assembly. Compared to the existing genomic DNA, a new DNA sequence is aligned to identify any similarities or differences between them to illustrate certain characteristics (Delcher, Phillippy, Carlton, & Salzberg, 2002). An essential part of an assembly is alignment, and it involves putting a massive number of reads of DNA, checking for regions where they align with one another, and eventually joining the puzzle together (Pevzner, Tang, & Waterman, 2001). This is the effort to rebuild the original genome, which is attempted by any assembler. Fig. 4.1 shows the whole protocol from DNA extraction step followed by sequencing to the draft genome assembly construction process to get back to the original genome.

4.1.2 Assembler technology: historical landscape

In the early 1980s, the very first sequence assembler was developed to join DNA fragments generated from sequencer (Shendure et al., 2017). TIGR, PHrap, and Cap3 were the first genome assemblers being used for viral and bacterial genomes. The technology for assembling DNA sequences has developed dramatically over time as the advancement in genomics

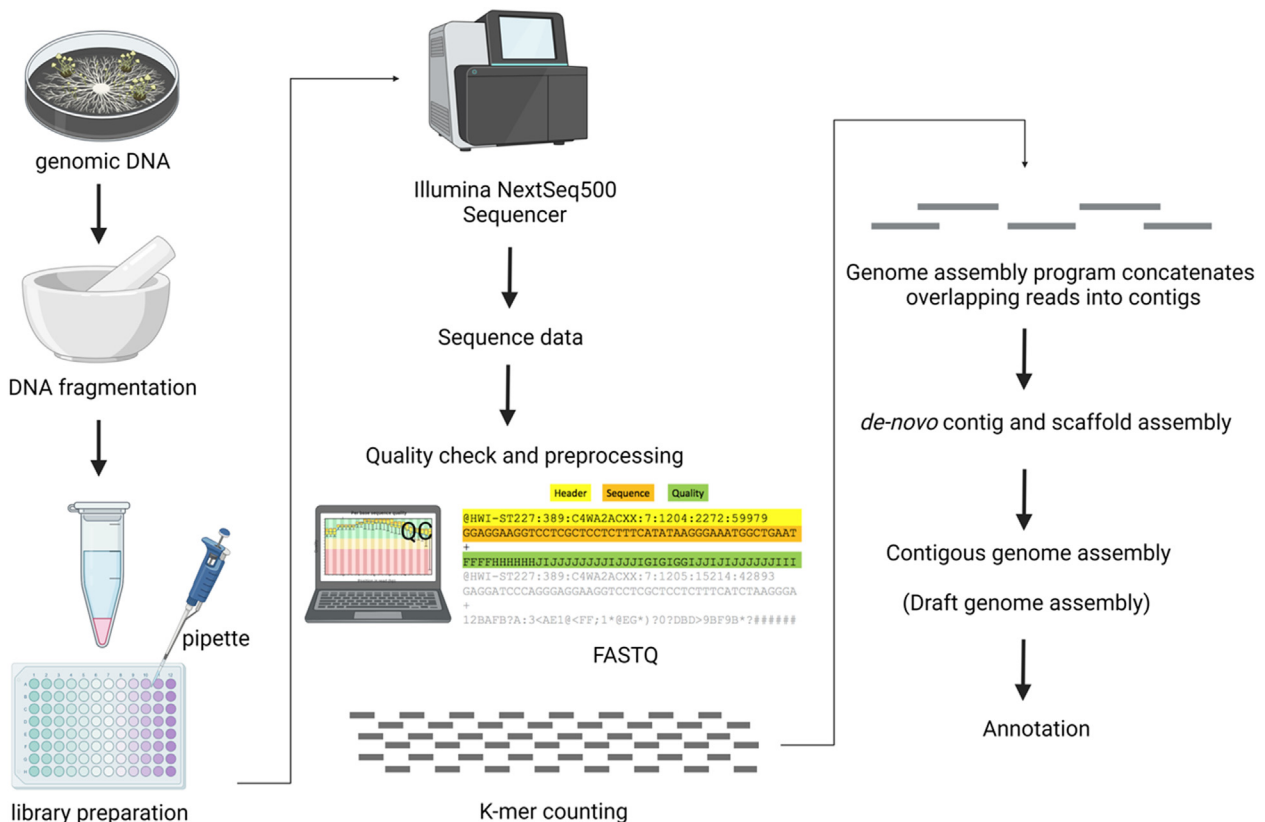


FIGURE 4.1 Schematic representation of sequencing and assembly process.

has needed further advanced techniques to handle the knowledge ongoing in the genomic research. To assemble the genome through huge amount of sequencing data, high-performance computing machines being used. Similar or identical fragments of DNA referred as repeats increased the assembling algorithms complexity with the computational time (Chin et al., 2013). In addition, it becomes hard to locate the sequencing data error if any small errors in the DNA fragments due to incorrect calibration of the sequencing machines (Kircher, 2012). Hence, with the time now new advancement being introduced to further enhance the DNA assembly process. Recent advances in the next-generation sequencing and parallel advancement of bioinformatics software together empowered the scientists to develop more way of genome assembly.

In the early days of DNA sequencing, the scientists were able to assemble only a few short length of sequences after weeks of lab work. The assemblers came into existence as variations in simple sequence alignment programs in the early 1990s. Scientists then faced the problem in 2000 while assembling the first large eukaryotic genome *Drosophila melanogaster* and the human genome and then established assemblers, such as Celera and Arachne assembler capable of handling 130 million to 3 billion base pair genomes. Following to this, scientists from several sequencing centers initiated an open source community to bring all the genome assembling technology developments under one open source platform known as AMOS and developed a large-scale assembly tool. By 2004–05, 454 Life Sciences had taken pyrosequencing toward its commercialization. This new method of sequencing produced reads far shorter than those of sequencing Sanger. The usage of this technology by genome centers was motivated by its much higher throughput and lower cost, which particularly pushed the development of genome assemblers that could use the read more effectively. Newbler 454 assembler was also came at the beginning of 2004. Then, the first freely available MIRA assembler in 2007 developed that could assemble 454 reads and Sanger reads. Later, Ion Torrent, Applied Biosystems SOLiD, PacBio SMRT, and Oxford Nanopore were launched the new technologies continued to emerge the new tools, which developed the Abyss, Spades, Canu, Falcon, etc., as freely accessible open source assembly programs. It allows research groups from the fields of genomics, genetics, agriculture, virology, medicine, forensic science, and microbiology to provide answers to many diverse problems. These are expected to contribute in the future to optimizing the accuracy and efficiency of genomic DNA assembly so that the findings could be used in realistic applications.

4.2 Genome assembly algorithms

Genome assembly algorithm is the collection of well-defined processes of constructing numbers of short sequence reads to original DNA sequences. Sequencing reads are aligned on each other and parts that overlap are found being combined to one stretch of sequence. Currently, two different methods of assembly algorithms are already in use that differ according to the complexity of sequencing data.

4.2.1 The overlap-layout-consensus/string graph assemblers

This algorithm recognizes intersects among combinations of reads to construct a graph of the connections between the sequencing reads (Li et al., 2012). The overlap-layout-consensus (OLC) is computational intensive approach where the complexity of computation increases with the total sequencing data used during assembling. Due to which this algorithm of assembly becomes inexcusable with the sequencers like Illumina where millions of short sequence reads will be needed for an assembly. After generating the graph, it visits through the each node using path called as Hamiltonian path, to construct the final assembly. The layout stage of the algorithm reduces the complexity of the preliminary graph by condensing the areas that unambiguously arose from the same genomic loci to the node where the line diverges with several potential paths. It is far from straight forward to decide such a path. This gives subgraphs to make contigs that can be described as unambiguously assembled unitig sequences that have high sequencing depth and linked to large numbers of other contigs. Now, the unitig is then paired with other unitigs to form a scaffolds sequence (Bresler, Sheehan, Chan, & Song, 2012). The last step of making the consensus process includes reading through the contiguous subgraphs and extracting the sequence of consensus for reads from each subgraph. Another string graph algorithm involves same overlap graph theory but slightly differs as it simplifies the graph by removing transitive edges that have redundant details (Chang et al., 2012).

4.2.2 De Bruijn graph assemblers

De Bruijn graph assemblers break down the reads into k-length subsequences (termed as k-mers) and used to build a graph. Since the graph nodes represent k-mers in this case, the edges denote neighboring k-mers that overlap with k-1 bases (Healy, 2010). The reads are not depicted in the graph directly and are represented by paths. Since this structure is based on the precise identity between k-mers. In graph, divergent paths are generated by sequencing errors that

reduce the length of the paths. Assemblers usually monitor each node's k-mer coverage, enabling the graph to be cleaned up by eliminating tips for low coverage (Heydari, Miclotte, Van de Peer, & Fostier, 2019). Usually, De Bruijn graph-based assemblers are very memory intensive, although various other methods are being used to minimize the use of memory (Chaisson & Pevzner, 2008).

4.3 Data preprocessing

4.3.1 Quality control of raw sequencing data

Before assembly, quality of the sequenced data, GC content, per base sequence quality score, per base N content, sequence length distribution, sequence duplication, overrepresented sequences, and content of k-mer and adapter sequences must be tested. Software such as FastQC provide basic statistics from the point of start. Primer and vector sequences are most likely to be present in the data obtained from library preparation and can be eliminated using easy scripts or software like Cutadapt (Martin, 2011). These sequences cause contamination; failure to eliminate such rich sequences of contaminants may interrupt the assembly process and can lead to chimeric and contaminated contigs' production. A short read aligner (BWA) is the best way to remove known vector contamination from the raw data (Li & Durbin, 2009).

4.3.2 Trimming and filtering of raw reads

In the trimming process, we remove mainly adapter sequences and low-quality reads resulting from PCR duplications (Fig. 4.2). These sequences should be removed from sequencing reads because they cause problems in downstream analysis. Adapter trimming requires only on the 3' end of reads because adapters are not present on the 5' end. It could be completed with various software and scripts. For several datasets, offline error correction by means of a k-mer count method (SOAPdenovo) may also be a useful alternative. ALLPATHS-LG is also another important tool, where the trimming and correction of error inside the assembly pipeline is carried out and raw reads are even required, without quality trimming as input (Gnerre et al., 2011). Some examples include FASTX-Toolkit (http://hannonlab.cshl.edu/fastx_toolkit/), Trimmomatic (Bolger, Lohse, & Usadel, 2014), and Cutadapt (Martin, 2011) tools that are freely accessible for the trimming of adapters and unpaired reads.

4.4 Genome assembly approaches: types of assembly

Genome assembly is a step that follows sequencing to reconstruct the genome from its reads (Pop, 2009). It refers to the process of placing the sequence of nucleotides into the correct order (Lu, Giordano, & Ning, 2016). Just as it is easy to assemble a picture puzzle, if you know what the image looks like, in the same way it is easy to assemble genome if one has good experience of ordering the sequence. In the human genome, genes fall on the chromosome at the same physical location, but repeated sequences may have different copies and variable numbers that make assembly complicated. Here are two types of assembly approaches that may be used: de novo and comparative reference-based assembly, described below in detail for a clear understanding between these two approaches. The way a picture puzzle is easy to assemble, if you know what the image looks like.

4.4.1 De novo assembly approach

De novo sequencing refers to the sequencing a new genome in which there is no reference genome sequence for alignment (Li et al., 2010). Simply, if reference genome sequence is not available, we will go to the de novo assembly (Zerbino & Birney, 2008). Based on the principle of OLC, information stored in scattered readings is used to create contiguous regions called "contigs" that are usually devoid of polymorphisms (Baker, 2012). Paired-end reads perform much better than single reads in the de novo assembly approach because it generates scaffolds. This assembly can be assessed based on various criteria, such as the number and size of available scaffolds and contigs and the fraction of reads that could be assembled (Tritt, Eisen, Facciotti, & Darling, 2012). A generally used metric for evaluating assembly quality is the contig and scaffold N50 value. An N50 contig is the smallest contig size, as the sum of the contigs of that size or longer constitutes at least 50% of total size of the contigs assembled (Makinen, Salmela, & Ylinen, 2012). The length of the smallest scaffold is the N50 scaffold size, such that the number of scaffolds of that size or longer makes up as a minimum 50% of the total size of all assembled scaffolds. Assemblers commonly used are based on the

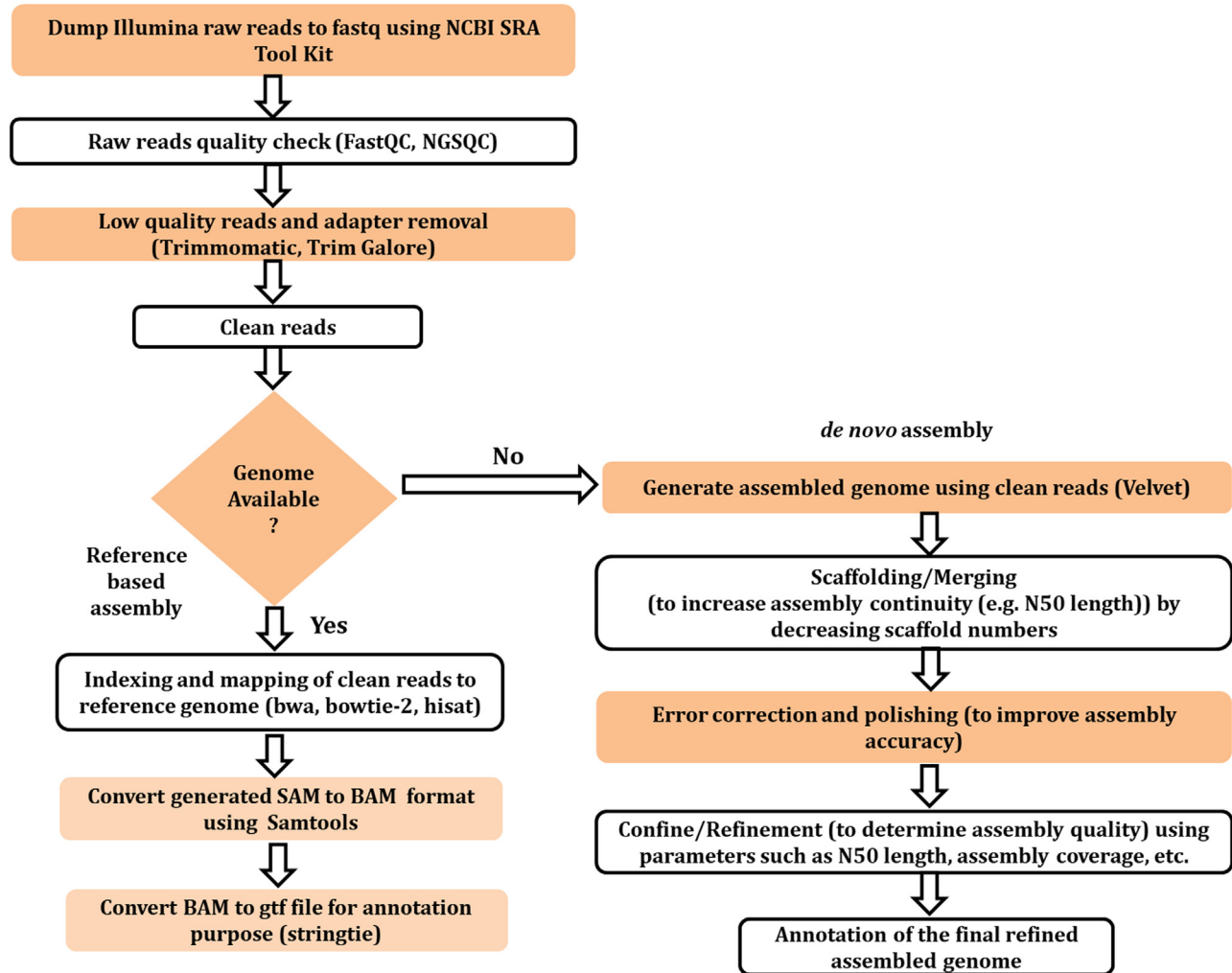


FIGURE 4.2 Showing steps of de novo and reference based assembly.

De Bruijn graph method, in which reads are split into subsequences called k-mers of length k (Kamal et al., 2017). The k-mers form the graph's nodes, which are associated when a k-1mer is shared among them. A large amount of computer memory (RAM) and specialized computer clusters are needed for this overall operation. Several incorrect recombinants are one of the key constraints of de novo assembly. De novo assembly refers to the use of short overlapping reads without reference genome sequence to obtain a complete genome sequence for a specific species (Li et al., 2010). This assembly method generates two distinct sequences: contigs and scaffolds (Boetzer, Henkel, Jansen, Butler, & Pirovano, 2011).

The de novo assembly approach divided into three methods viz., the greedy graph method, De Bruijn graph method, and overlap-layout consensus method (Li et al., 2012). Most elongated overlapping regions between different reads are searched by the above methods; then, these regions are constantly combined to form a contig. The assembly tools that use greedy graph method are SHARCGS, SSAKE, and VCAKE (Lin et al., 2011). They adopt a simple but efficient method in which the assembler greedily joins the reads that are most identical to each other. Although the assembler could easily be confused by complex repeats that lead to misassembled contigs since only local information is recognized at each phase. Assembly tools, such as Newbler, Celera assembler, and Edenaare, are based on OLC methods (Miller, Koren, & Sutton, 2010; Zerbino & Birney, 2008). There are three most important software, which use De Bruijn graph to produce assemblies: SPAdes (Bankevich et al., 2012), Unicycler (Wick, Judd, Gorrie, & Holt, 2017), and Velvet (Zerbino & Birney, 2008). Velvet is designed for de novo genome assembly, which builds De Bruijn graphs using short reads and removes errors from graphs (Zerbino & Birney, 2008). SPAdes is designed to assemble reads of small genomes and single cell sequencing. It takes paired-end, mate pairs, and unpaired reads as input otherwise it

causes error. Unicycler resolves bacterial genome assemblies from both sequencing reads (short and long). The assembly produced by unicycler is cost effective and accurate.

4.4.2 Reference-based assembly approach

Comparative mapping or reference-based assembly used for those organisms that have an existing assembled genome sequence, which is similar to another organism, can be used as a reference (English et al., 2012). In reference-based assembly, approach comprises mapping of each read to a reference genome to identify variations on genetic level, such as SNPs, copy number variants, indels (insertions and deletions), genome-wide association studies, and haplotypes from genome assemblies (Huang & Han, 2014). Bowtie and BWA are memory-efficient and ultrafast short-read aligners that help in assembly and mapping. Bowtie speedily aligns large dataset of short reads (35-bp reads) to a reference genome sequence, at a rate of above 25 million reads/h. For reads longer than 50 bp, Bowtie2 is used that is usually more sensitive and faster and uses less memory than Bowtie (Langmead, Trapnell, Pop, & Salzberg, 2009). The best performing aligner is BWA, which achieves the highest alignment rate.

4.4.3 Hybrid assembly approach

During starting of genome sequencing project, there was a clear trend away from the traditional sequencing method, that is, Sanger sequencing to the technologies, such as Roche 454 sequencing, SOLiD, and Illumina HiSeq. Recently there has been progress in the production of long reads at high-throughput rate; so various technologies, namely Ion Torrent (~500 bp), Pacific Biosciences (8–15 kb and up to 40–70 kb), Illumina Moleculo (~10 kb), and Nanopore, are entering in the marketplace, and we are waiting for wide variety of read lengths (Ekblom & Wolf, 2014; Peng et al., 2014). Finishing complete chromosomes generally requires the use of several sequencing technologies and protocols of hybrid assembly. You will often see short-read technology combined with long-read technology to generate fully finished genomes. Using the hybrid approach, in Fig. 4.3, contigs were assembled with high-quality error-corrected Illumina and Pacbio reads using hybrid SPAdes, which is based on the De Bruijn graph approach with the highest accuracy.

4.4.4 Meta-assembly approach

Building a draft genome is an iterative method that requires optimization of parameters. It is recommended that more than one form of assembly with different experiments and sequencing approach can be used to merge them to improve the contiguity and fill the gaps. This approach based on the reconciliation technique (Fig. 4.4) includes (1) merging of two misassemblies, (2) discovering matches, mismatches, and other errors in sequencing, and (3) closing gaps using mate-pair, pair-end libraries (Marla et al., 2020). Metassembler tool employed to detect gaps and filled those iteratively

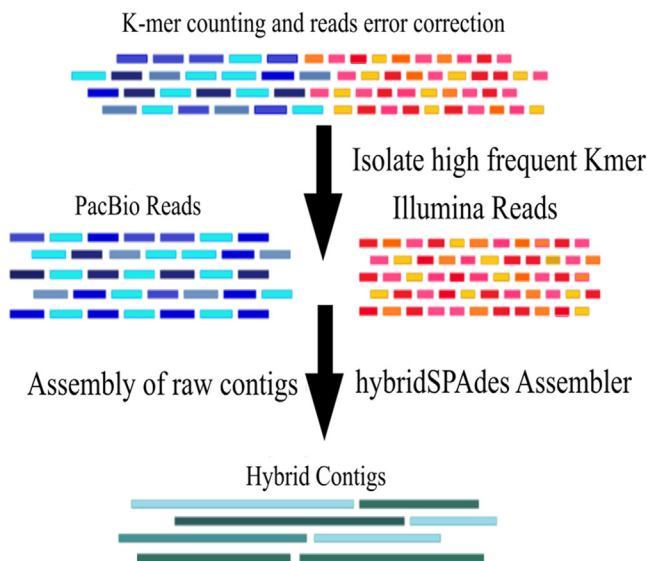


FIGURE 4.3 Hybrid assembly approach (Mishra et al., 2019).

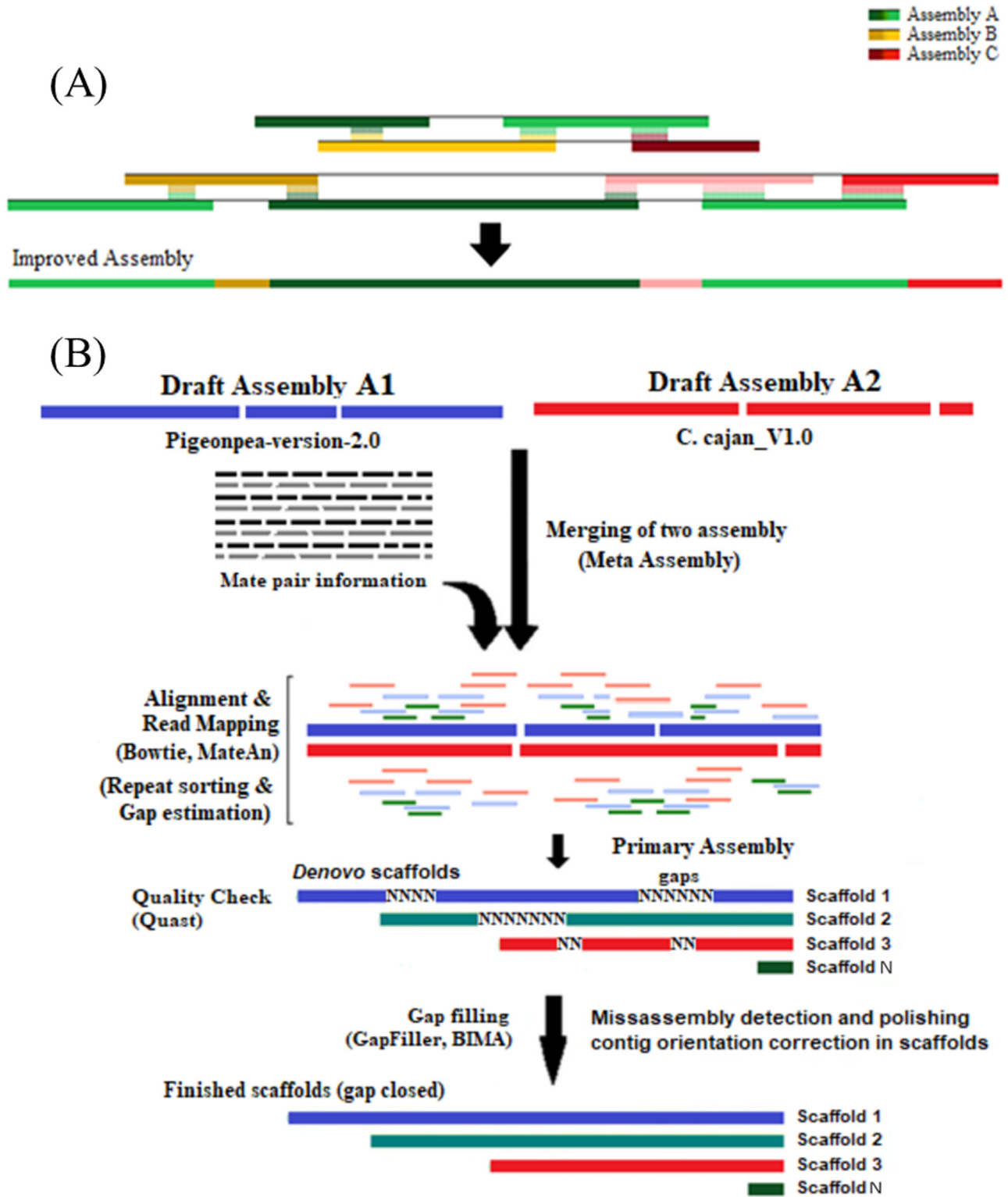


FIGURE 4.4 Experimental framework depicting merging steps for genomes. (A) Metassembler tools used to identify overlaps between all three assemblies representing overlapping aligned and unaligned regions between two contigs. (B) High confidence overlaps identified by metassembler tools used to join each contig and show the resulted improved assembly (Marla et al., 2020).

selecting right-sided inserts from local mate-pair and pair-end libraries. Mis-assemblies are not accounted for in N50 parameters due to incorrect read orientation and low complexity regions. For this, amosvalidate tool is recommended, which incorporates several validation procedures (Kasibhatla, Waman, Kale, & Kulkarni-Kale, 2016).

4.5 Tools and software for genome assembly

Several tools are available for assembling long-read sequences generated through high-throughput sequencing platforms. To guide researchers in the field of bioinformatics, selection of suitable sequencing tools is very important. Several parameters that help to select tools are based on their performance, efficiency, reliability, type of assembly, etc. Based on various properties, list of software used in genome assembly is given below (Table 4.1).

4.5.1 Genome finishing/polishing

Genome finishing, which means the order of all nucleotide bases, has been correctly resolved. Even for simple genomes, this is extremely difficult. Filling gaps in supercontigs is an important step in genome finishing. Finished assembly has reduced the number of gaps than reported draft assemblies and has improved genome coverage. To identify specific trait-related functional genes, quality of the finished assembly can be evaluated using various quality metrics. In Fig. 4.5, the authors showed the genome-finishing steps, where raw reads were realigned to the scaffolds to fill the high N contents in the genome assembly. There are so many tools available for genome finishing, viz. Gapcloser (Xu et al., 2019), FGAP (Piro et al., 2014), IMAGE (Tsai, Otto, & Berriman, 2010), GapFiller (Nadalin, Vezzi, & Policriti, 2012), and GFinisher (Guizelini et al., 2016).

TABLE 4.1 List of whole genome assembly tools along with their description.

Software/tools	Description with URL	References
A5	It is an automatic pipeline for genome assembly, which includes five steps: cleaning reads, assemble error corrected reads, scaffolding, scaffold validation, and scaffold assembly (https://sourceforge.net/projects/ngopt/)	Tritt et al. (2012)
ABYSS	It is a de novo genome assembly tool (https://www.bcgsc.ca/resources/software/abyss)	Simpson et al. (2009)
ALE-Assembly	It is used for evaluating accuracy of assemblies (https://bioinformaticshome.com/tools/wga/descriptions/ALE-Assembly.html)	Clark, Egan, Frazier, and Wang (2013)
ALLPATHS	It is used for genome assembly, which is applicable for all types of sequences (short and long reads) (ftp://ftp.broadinstitute.org/pub/crd/ALLPATHS/Release-2-0/)	Butler et al. (2008)
Allpaths-LG	It is an updated version of ALLPATHS, which is used for assembling both small and large genomes (https://www.broadinstitute.org/computational-rd/computational-research-and-development)	Gnerre et al. (2011)
aTRAM	It is used for assembling targeted genes from single library (https://github.com/juliema/aTRAM)	Allen et al. (2017)
CANU	It is a tool to assemble long reads of both platforms PacBio or Oxford Nanopore (https://github.com/marbl/canu)	Koren et al. (2012)
CAR	It is used to rearrange contigs based on reference sequence. (http://lu168.cs.nthu.edu.tw/CAR/index.php)	Lu, Chen, Huang, and Chiu (2014)
Celera	It is a key assembler because it has the capability to assemble the genomes of multicellular organisms (https://sourceforge.net/projects/wgs-assembler/)	Myers et al. (2000)
dnaasm	It is used for assembling tandem repeats (http://dnaasm.sourceforge.net/)	Kusmirek and Nowak (2018)
Edena	It is used for de novo genome assembly (http://www.genomic.ch/edena.php)	Hernandez, François, Farinelli, Østerås, and Schrenzel (2008)

(Continued)

TABLE 4.1 (Continued)

Software/ tools	Description with URL	References
ELOPER	During assembly, it is used to preprocess paired-end short reads for better performance. (https://sourceforge.net/projects/eloper/)	Silver, Ben-Elazar, Bogoslavsky, and Yanai (2013)
Enly	It is used for closing gaps in genome assembly (https://sourceforge.net/projects/enly/)	Fondi et al. (2014)
EULER	It is used for de novo genome assembly, which is based on De Bruijn algorithm (http://web.archive.org/web/20110706023620/http://nbcrc.sdsc.edu:80/euler/euler2_dl/)	Pevzner et al. (2001)
FALCON	It is used for de novo assembly of long PacBio reads (https://github.com/PacificBiosciences/falcon)	Chin et al. (2016)
GAM-NGS	It is used to merge two or more assemblies (https://github.com/vice87/gam-ngs)	Vicedomini, Vezzi, Scalabrin, Arvestad, and Policriti (2013)
GAML	It is used for genome assembly based on maximum likelihood (http://compbio.fmph.uniba.sk/gaml/)	Boza, Brejova, and Vinar (2015)
HapCol	It is used for assembling sequences into haplotypes from population-sampled data (http://hapcol.algolab.eu/)	Pirola et al. (2016)
Kermit	It uses linkage maps to guide genome assembly (https://github.com/rikuu/kermit)	Walve, Rastas, and Salmela (2019)
Kourami	It is used for assembling HLA haplotypes (https://github.com/Kingsford-Group/kourami)	Lee and Kingsford (2018)
Mapsembler	It is used for targeted assembly of particular genomic locus (http://colibread.inria.fr/software/mapsembler2/)	Peterlongo and Chikhi (2012)
MaSuRCA	It is a hybrid approach which combines De Bruijn graph and overlap-based assembly strategies (https://github.com/alekseyzimin/masurca)	Zimin et al. (2013)
MECAT	It is used for de novo assembly of long sequence reads (https://github.com/xiaochuanle/MECAT)	Xiao et al. (2017)
miniasm	It is used for de novo assembly of long reads either from PacBio or Oxford Nanopore platforms (https://github.com/lh3/miniasm)	Li (2016)
PAGIT	It is used to improve the quality of assembled genomes (https://www.sanger.ac.uk/science/tools/pagit)	Swain et al. (2012)
PASHA	It is used for assembling genomes based on short reads (http://pasha.sourceforge.net/)	Liu et al. (2012)
Phusion	It is used for de novo genome assembly (https://www.sanger.ac.uk/science/tools/phusion)	Mullikin and Ning (2003)
QUAST	It is used for genome assembly evaluation (http://quast.sourceforge.net/)	Gurevich, Saveliev, Vyahhi, and Tesler (2013)
SGA	It is used for de novo genome assembly (https://github.com/jts/sga)	Simpson and Durbin (2012)
SMRT	It is used for calling SNPs and assembling haplotypes based on PacBio (long) reads (https://github.com/guofeiileen/SMRT/wiki/Software)	Guo, Wang, and Wang (2018)
SOAPdenovo2	It is used for de novo genome assembly which is suitable for short reads (https://sourceforge.net/projects/soapdenovo2/)	Luo et al. (2012)
Velvet	It is used for de novo assembly based on De Bruijn graphs (https://github.com/dzerbino/velvet)	Zerbino and Birney (2008)

Source: <https://bioinformatics.home.com/tools/wga/wga.html>. Post assembly requisites.

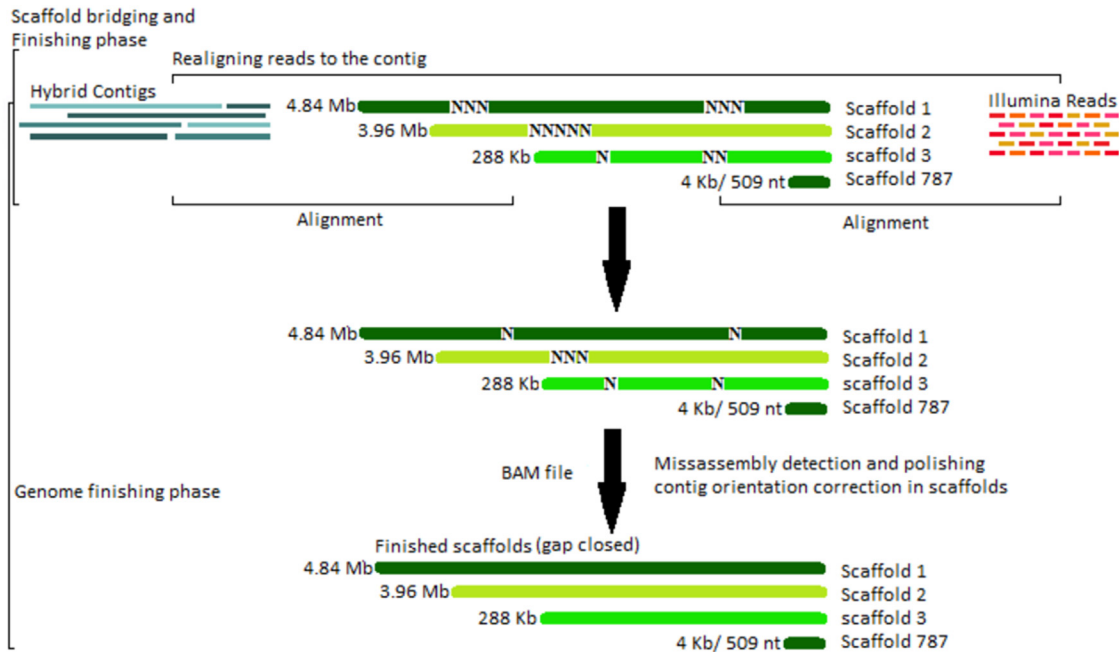


FIGURE 4.5 The sequence finishing process of an assembly depicting problems due to incomplete coverage or poor-quality data (Mishra et al., 2019).

4.5.2 Assembly quality assessment and validation

After successful assembly, users want to evaluate the quality of assemblies through various methods. Each draft assembly of genome builds a hypothesis of the correct original genome sequence, as discussed above, and assessing the accuracy remains a challenge in the absence of knowing the facts. There are a number of metrics available that reflect various elements of the assembly (Bradnam et al., 2013). *C*-value or *k*-mer frequency-based methods may be used to estimate the size of the genome (Ekblom & Wolf, 2014). The N50 statistic is another common metric for evaluating assembly contiguity. According to definition, N50 is alike mean or median of assembled contig (contig N50) or scaffold (scaffold N50) lengths, given greater weight to longer sequences. The first step towards the successful annotation of any genome is to determine its assembly. After that, it is ready for annotation (Yandell & Ence, 2012). Average gap size of a scaffold and gaps per scaffold is another important statistical parameter used to assess the quality of assembly. However, a high-quality draft assembly can be used for genome annotation.

However, N50 scaffold length assembly represents an enough length for a gene to perform annotation (Yandell & Ence, 2012). CEGMA screens an assembly against a set of universal eukaryotic single-copy genes and determines the percentage of each gene lying on a particular scaffold. The aim of genome sequencing is to find the high-quality draft assembly. If the length of an assembly is very short, then additional shotgun sequencing can be performed (Yandell & Ence, 2012). The BUSCO tool universally identify single-copy orthologs, which often have core metabolic genes that exist in nearly all (> 90%) species in each phylogenetic clade, that appear in a sequenced sample file. A resulting run reports the number of BUSCOs discovered in a given sample that is fully intact (complete), partially found (fragmented), or missing from the sample entirely. BUSCO is often used to measure completeness by adding the complete and fragmented percentages to ensure that a newly assembled genome/transcriptome has a majority of the nearly universal core genes. As input, BUSCO requires a FASTA file.

4.6 Pitfall in genome assemblies

4.6.1 DNA quality

Prior to sequencing, inherent properties of the genome are not the only concern. Other aspects that require careful preparation are also available. One such factor that is of utmost significance is the extraction of intact, high-quality, and nondegraded genomic DNA (Panova et al., 2016). There are often over contaminants in DNA extract that produce either from the beginning material or from the DNA extraction process. Metabolites, polysaccharides, polyphenols,

proteins, and pigments are one of the major contaminants. For example, bacterial and plant samples may have high polysaccharides' level. So, there are more chances to contaminate with polysaccharides. Similarly, insect samples can contaminate with proteins, polysaccharides, and pigments. Altogether, these contaminants can damage the effectiveness of library preparation in any technique, but this is particularly correct for the preparation of PCR-free libraries and Illumina Mate Pair libraries. A complete genome requires about 1 mg of DNA as a preliminary material for sequencing (approximately 6 μg for short insert, approximately 40 μg for 2–10 kb, approximately 60 μg for more than 20 kb libraries). A significant volume of high-quality genomic DNA of the target organism must be collected before engaging genome sequencing (Eklom & Wolf, 2014). Before submitting a sample of DNA, the integrity of the sample must be tested on a gel of high resolution using pulsed-field gel electrophoresis.

4.6.2 Library preparation

When selecting the required raw read depth, we should know that most technologies currently require multiple PCR steps, which can give rise to a nonnegligible number of duplicate reads (Eklom & Wolf, 2014). As duplicated reads, there is no additional benefit and duplicate artefacts may affect the validation of coverage-based quality, so they must be removed before assembly. Generally, duplicates constitute a few percent of short-insert libraries (less than 500 bp) but, for long-insert libraries (more than 10 kb), can cross over 95% (Salk, Schmitt, & Loeb, 2018). Another main issue relates to what sizes of inserts to use. In general, a good mix of sizes range from 0.2 to 40 kb is suggested, with smaller libraries being sequenced too much higher depths (Gnerre et al., 2011). Insert sizes of more than 20 kb allow a major variation between the assembly's final contiguity and scaffold size. The primary step of next-generation sequencing is library preparation. This allows the DNA to adhere the flow cell of sequencing and helpful in identifying samples. For library preparation, two methods are commonly used, that is, library preparation based on tagmentation and library preparation based on ligation.

4.6.3 Data quality

Sequence data can also contain errors. The depth of sequencing plays a significant role in determining the quality of assembly (Gnerre et al., 2011). Current sequencing technology does not produce 100% correct sequences all the time. Sometimes, it shows errors based on the quality of extracted DNA. The data that comes from the wet lab for sequencing are most prone to error (Dohm, Lottaz, Borodina, & Himmelbauer, 2008). However, variable sequence quality is not restricted to the ending of the reads, as secondary structure effects or the compressions of GC-rich region peaks can occasionally be poorly resolved in the middle of the sequence. Therefore the base caller may not recognize the correct base or the correct number of bases, making sequencing data “noisy” (McGrath, 2007). Assembly programs are influenced by noisy data when trying to find overlaps between fragments, as single base differences could prevent otherwise identical sequences from overlapping.

Assembly programs need to balance between being able to recognize true overlaps in the presence of sequencing errors while also not increasing the amount of repeat-induced overlaps (Koren et al., 2012). This method calculates the error probability for each base call and is implemented in the program Phred. Since its introduction in 1998, Phred quality value has become the industry standard and it remains the most popularly used method of base-call error estimation. Phred quality values range from 0 to 99, and increasing values indicate increasing quality. The Phred quality score value of 10 corresponds to a 90% chance of accuracy, or 1 in 10 chances that the base call is being incorrect. The standard that is mostly expected is a Phred quality of 20, Q20, that corresponds to a predicted error rate of 1% or a 99% chance of the base call being correct.

4.6.4 Repetitive DNA

As previously noted, while it would be expected that a single contig representing the entire genome would be restored after genome assembly, this is seldom the case, even for simple genomes due to several complicating factors with the underlying raw data (Miller et al., 2010). The main complicating factors affecting genome assemblies are repetitive sequences and biological artifacts resulting errors in genome assembly. Ideally, the assembly program should differentiate among different copies of the repetitive sequences and place them in their right genomic location. However this is not the case and the problem of misassembled genomes resulting from repetitive sequence is widespread and all assembly tools are affected (Salzberg & Yorke, 2005). Longer repeats are problematic to assemble because even though they originate from different locations in the genome, the assembly program cannot distinguish between the different

repeated copies, the reads are all assembled together which tends to result in collapse the assembly at that point. This is especially true in tandem repeats, resulting in a (apparent) very deep coverage of this region compared to the rest of the genome. In this example, as both copies of the repeat are assembled together resulting in stacking at that point (Miller et al., 2010), the intervening reads in the region between the repeats can only be assembled together and can't be included in the assembly. The stacking results in an incorrect genome reconstruction, and the correct reconstruction must be resolved by labor-intensive and time-consuming finishing processes (McGrath, 2007).

Identical repeats are the most problematic as it is impossible to distinguish between copies of the repetitive element. Repeats can be of a number of different categories, such as microsatellites, low-complexity DNA, transposons, or retro-transposons, such as short interspersed nuclear elements (e.g., Alu repeats, 300 bp in length) and long interspersed nuclear elements (LINEs of 500–5000 bp in length), or LTR retrotransposons, such as long-terminal repeats of approximately 700 bases in length. Other classes of repetitive sequence that can cause problems for an assembly program include gene duplication where genes duplicate and then diverge in sequence, or long-segmental duplications that can be very long and have very similar copies of a very long portion of the genome. Mis-assemblies of repetitive DNA containing regions can therefore result also in the excision of a repeat in particular genome locations, as well as erroneous genome rearrangements, in addition to collapsed areas of the genome. Techniques for repairing regions of the genome that contain repetitive DNA are labor intensive and depend on finding differences between the sequencing reads that are aligned together at a particular point.

4.7 A mathematical calculation for depth/coverage

Lander and Waterman (1988) helped formulate a mathematical framework for estimating various statistical parameters associated with sequencing. Their intention was to provide a model for the physical mapping projects that were underway at the time. While it was not intended to model shotgun sequencing they have since been used in this way, and these statistical measures were widely used when planning the public human genome sequencing project. The Lander–Waterman model has been used to determine the contig landscape of a randomly fragmented genome given the length of the genome, the average read length, the number of fragments studied, and the minimum overlap used for the sequence assembly. It is commonly used to determine the number of sequencing reads required for a given shotgun genome coverage, and the expected number of gaps in an assembly, as functions of the number of reads sequenced. For the calculation of depth, SAMtools depth computes the read depth at each position or region by using the sorted BAM files as an input. Lander and Waterman (1988) method is used for calculating coverage. The general formula of coverage is:

$$\text{coverage}(C) = LN/G$$

where C = coverage; G = haploid genome length; L = read length, and N = number of reads.

4.8 Genome annotation

In genome annotation, we simply identify coding (exons), noncoding (intron) regions and locations of genes present in genome sequence and determine what those genes exactly do (Avashthi et al., 2018; Stein, 2001; Tiwari et al., 2016). Genome annotation has three key objectives: (1) to identify noncoding parts of the genome, (2) to find the function (gene) part of a genome, and (3) to know the functional role of genomic elements. Genome annotation requires an automated system to copy, paste, run, visualize, and analyze the BLAST results (Fig. 4.6). Final decision related to genome annotation for finding gene structure or function is made on the basis of manual analysis of generated results by the experts (Avashthi et al., 2018). Numerous efforts have been made for automation that have data processing and decision-making capabilities.

To provide the complete information of a genome sequence, we need to interpret it biologically that can range from gene sequence to functional annotation, for example, gene ontology (GO) terms or Kyoto Encyclopedia of Genes and Genomes (KEGG) pathways (Kanehisa & Goto, 2000). Annotating genes in newly sequenced organism and preexisting annotated gene sequences are required, providing reference to annotate gene for individual organism (Yandell & Ence, 2012). Before gene prediction, removal of repetitive sequences, transposable elements, and low-complexity regions is important. Sometimes, as repeats are repeatedly found across species, it is suitable to build an organism-specific library of repeat through such tools, for example, Repeat Modeler, Repeat Masker, and Repeat Explorer (Avashthi et al., 2020; Novak, Neumann, Pech, Steinhaisl, & Macas, 2013). Genome annotation is of two type structural and functional annotations. For complete genome annotation require considerable efforts and bioinformatics proficiency.

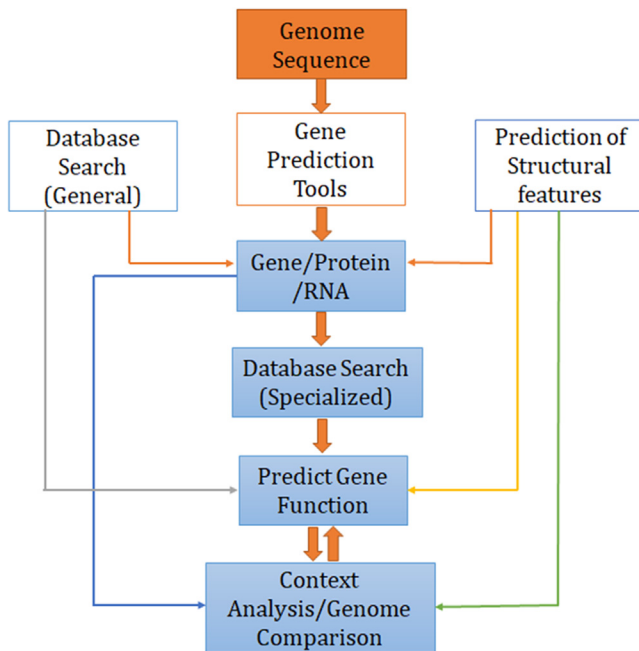


FIGURE 4.6 Flow chart of genome annotation.

4.8.1 Structural annotation

A genome sequence has much more value if we can annotate the distinct features, such as regulatory motifs, promoters, start/stop codon, introns, exons, splice sites, transposons, and coding sequences. Annotations of those elements show the functions for these different regions called structural annotation (Avashthi et al., 2020). These annotations are actually helpful for those users who are working with genomic sequence. Let us assume that we link a particular disease to a region of a genome. If we have a bad annotation or no information about the annotation, we will not know which genes are close to that region even if we have the gene. Overall, if we have a good quality annotation, then it gives an understanding of the field of study.

We can base also structural annotations on experimental data, for example, EST, or we can do in silico analysis in sequence by ourselves. Subsequent annotations that begin only with the sequence to be annotated are generally referred to as ab initio annotations. There are various bioinformatics tools, which are available to analyze and to predict gene structure (promoters, exons, and introns), alternative splicing, coded proteins, and many more. These tools include AUGUSTUS, MAKER, Glimmer, Gene Structure Display Server, Translate, Transeq, Backtranseq, ORF Finder, Promoter2.0, TFBIND, etc. (Avashthi et al., 2020). Few of them are discussed below:

4.8.1.1 AUGUSTUS

AUGUSTUS (<http://bioinf.uni-greifswald.de/webaugustus/prediction/create>) is a software that is used to predict genes in eukaryotic-sequenced genome. This program is based on probabilistic model called Hidden Markov Model (Stanke et al., 2006). For short files, it is available online and can be used as web server. For large datasets, it can be downloaded and run locally after installation. The performance of this tool has been assessed using large-scale sequence data of human and *Drosophila*. The accuracy for long input sequences is higher to that of existing ab initio gene prediction tools. It provides gene's protein coding regions and protein sequences of the predicted genes.

4.8.1.2 MAKER

MAKER is a simple and convenient structural annotation pipeline. It is a combination of various tools, such as AUGUSTUS, BLAST, Exonerate, RepeatMasker, and SNAP. The purpose of this pipeline is to annotate smaller prokaryotic and eukaryotic genome independently and create genome databases (Campbell et al., 2014). It identifies repeats and aligns expressed sequence tags and proteins to a specific genome. The outcomes obtained from MAKER can be visualized easily using GBrowse, which shows predicted gene structure via above-mentioned inbuilt separate programs and provides final structure of gene predicted by MAKER.

4.8.2 Functional annotation

It is the process of collecting biological information about gene or protein, which describes different aliases, molecular function (biochemical/biological), molecular interaction, subcellular localization, domain, signal peptides, glycosylation sites, binding sites, low complexity regions, expression, biological activity, and identity within the organism. For this, we use various tools that run on the particular genome and predict the function of gene. BLAST is one of the basic annotation tools that find similarity based on homology and known structural annotation of gene and protein sequences. Nonetheless, more information regarding biological functions is added to the annotation system nowadays. The added information permits hand-operated annotation to differentiate genes or proteins, which have the similar annotation. With many sequenced genomes, computational annotation approaches are increasingly important to characterize genes and proteins from their sequence. For functional annotation, we can use several tools, such as Blast2GO, Blast + , KEGG, TargetP, STRING, Pfam, BLAST, InterProScan, Diamond, and SignalP (Maurya et al., 2020). Few of them are discussed below.

4.8.2.1 KEGG

The KEGG database (<https://www.genome.jp/kegg/kegg1a.html>) is developed for the purpose of revealing cellular functions from gene datasets in the complete genomes. The KEGG pathways are manually drawn network diagrams, which show interactions between metabolic, signaling, and other molecules. It is an integrated database comprising 18 databases. These are mainly categorized into four categories, such as chemical, genomic, systems, and health information. These categories can be differentiated using color coding of web pages. One more interesting tool of KEGGs is BlastKOALA, which is used for the annotation of high-quality genomes.

4.8.2.2 STRING

STRING refers to search tool for recurring instances of neighboring genes (<http://www.bork.embl-heidelberg.de/STRING>). It is a tool to display repeated occurrence of query genes on the genome. For this, FASTA sequence or the name of gene is taken as an input and the search starts from a single protein sequence. Entered protein sequence is compared against protein database and identifies orthologs using the Smith–Waterman algorithm.

4.8.2.3 Clusters of Orthologous Groups

The Clusters of Orthologous Groups (COGs; <http://www.ncbi.nlm.nih.gov/COG>) is a database that is designed by classifying proteins from fully sequenced genomes based on the orthology concept. It comprises 2091 COGs, which include gene products from bacterial and archaeal genomes. It shows different types of orthologous relationship, such as one to one, one to many, and many to many. In this database, genes are identically colored if their product belongs to the same COG.

4.8.2.4 Gene ontology

GO annotation describes the function of a particular gene. For GO annotation, Blast2GO tool is widely used. It allows automatic high-throughput sequence analysis (Conesa et al., 2005). It uses BLAST to find homologs and extract GO terms for each input sequence and provide functional annotation in three categories, viz. biological process, cellular component, and molecular function. This tool is very helpful to understand the physiology of multiple genes and to assess functional variations among group of sequences.

4.9 Application and future prospects of genome assembly

Draft genome sequences of various organisms are now being generated at an ever-increasing rate. Researchers publishing the draft genome to traditional databases, such as Ensembl, Phytozome, UCSC Genome Browser, Wellcome Trust Sanger Institute, International Wheat Genome Sequencing Consortium, and the genomic databases of National Center for Biotechnology Information (NCBI). These sources also provide curated data as well as raw reads of incoming genome of various organisms. NCBI already offers the opportunity to upload user-generated annotation, draft genome sequences, and meta-data in the form of BioProject. All available raw data and assembly uploaded to databases will permit other users to refine the assembly and their annotation. These assemblies and annotations will also be helpful to develop biotic and abiotic stress tolerance varieties of cereal crops and cure human from various diseases by identifying targets.

4.10 Conclusion

In this chapter, we have discussed about the assembly and annotation approaches that are used after sequencing of particular organism's genome. Here, we have tried to provide detailed information regarding genome assembly, pre- and postrequisites of assembly, pitfalls of assembly and further bioinformatics analysis in the form of structural and functional annotation using various tools. New assembly methods are being developed to arrange and interpret the original genome because of the ever-improving sequencing technologies. Therefore it is important to familiarize the reader with the updated data. A great opportunity for researchers to learn about the intrinsic characteristics of genome in terms of quality assessment will also be helpful for identifying candidate genes responsible for various diseases in plants and animals.

Conflict of interest

The authors declare that they have no conflict of interest.

References

- Allen, J. M., Boyd, B., Nguyen, N. P., Vachaspati, P., Warnow, T., Huang, D. I., Pittendrigh, B. R. (2017). Phylogenomics from whole genome sequences using aTRAM. *Systematic Biology*, 66(5), 786–798.
- Avashthi, H., Pathak, R. K., Pandey, N., Arora, S., Mishra, A. K., Gupta, V. K., Kumar, A. (2018). Transcriptome-wide identification of genes involved in ascorbate–glutathione cycle (Halliwell–Asada pathway) and related pathway for elucidating its role in antioxidative potential in finger millet (*Eleusinecoracana* (L.)). *3 Biotech*, 8(12), 499.
- Avashthi, H., Pathak, R. K., Gaur, V. S., Singh, S., Gupta, V. K., Ramekte, P. W., & Kumar, A. (2020). Comparative analysis of ROS-scavenging gene families in finger millet, rice, sorghum, and foxtail millet revealed potential targets for antioxidant activity and drought tolerance improvement. *NetMAHIB*, 9(1), 33.
- Baker, M. (2012). De novo genome assembly: What every biologist should know. *Nature Methods*, 9(4), 333–337.
- Bankevich, A., Nurk, S., Antipov, D., Gurevich, A. A., Dvorkin, M., Kulikov, A. S., Pyshkin, A. V. (2012). SPAdes: A new genome assembly algorithm and its applications to single-cell sequencing. *Journal of Computational Biology*, 19(5), 455–477.
- Batzoglou, S., Jaffe, D. B., Stanley, K., Butler, J., Gnerre, S., Mauceli, E., Lander, E. S. (2002). ARACHNE: A whole-genome shotgun assembler. *Genome Research*, 12(1), 177–189.
- Bayer, P. E., Hurgobin, B., Golicz, A. A., Chan, C. K. K., Yuan, Y., Lee, H., Zou, J. (2017). Assembly and comparison of two closely related *Brassica napus* genomes. *Plant Biotechnology Journal*, 15(12), 1602–1610.
- Boetzer, M., Henkel, C. V., Jansen, H. J., Butler, D., & Pirovano, W. (2011). Scaffolding pre-assembled contigs using SSPACE. *Bioinformatics (Oxford, England)*, 27(4), 578–579.
- Bolger, A. M., Lohse, M., & Usadel, B. (2014). Trimmomatic: A flexible trimmer for Illumina sequence data. *Bioinformatics (Oxford, England)*, 30(15), 2114–2120.
- Boza, V., Brejova, B., & Vinar, T. (2015). GAML: Genome assembly by maximum likelihood. *Algorithms for Molecular Biology*, 10(1), 18.
- Bradnam, K. R., Fass, J. N., Alexandrov, A., Baranay, P., Bechner, M., Birol, I., Chitsaz, H. (2013). Assemblathon 2: Evaluating de novo methods of genome assembly in three vertebrate species. *GigaScience*, 2(1), 2047.
- Bresler, M. A., Sheehan, S., Chan, A. H., & Song, Y. S. (2012). Telescope: De novo assembly of highly repetitive regions. *Bioinformatics (Oxford, England)*, 28(18), i311–i317.
- Butler, J., MacCallum, I., Kleber, M., Shlyakhter, I. A., Belmonte, M. K., Lander, E. S., Jaffe, D. B. (2008). ALLPATHS: De novo assembly of whole-genome shotgun microreads. *Genome Research*, 18(5), 810–820.
- Catchen, J., Amores, A., & Bassham, S. (2020). Chromonomer: A tool set for repairing and enhancing assembled genomes through integration of genetic maps and conserved syntenies. bioRxiv.
- Campbell, M. S., Law, M., Holt, C., Stein, J. C., Moghe, G. D., Hufnagel, D. E., Lei, J., Achawanantakun, R., Jiao, D., Lawrence, C. J., & Ware, D. (2014). MAKER-P: A tool kit for the rapid creation, management, and quality control of plant genome annotations. *Plant Physiology*, 164(2), 513–524.
- Chaisson, M. J., & Pevzner, P. A. (2008). Short read fragment assembly of bacterial genomes. *Genome Research*, 18(2), 324–330.
- Chang, Y. J., Chen, C. C., Chen, C. L., & Ho, J. M. (2012). A de novo next generation genomic sequence assembler based on string graph and MapReduce cloud computing framework. *BMC Genomics*, 13(S7), S28.
- Chin, C. S., Alexander, D. H., Marks, P., Klammer, A. A., Drake, J., Heiner, C., Turner, S. W. (2013). Nonhybrid, finished microbial genome assemblies from long-read SMRT sequencing data. *Nature Methods*, 10(6), 563–569.
- Chin, C. S., Peluso, P., Sedlazeck, F. J., Nattestad, M., Concepcion, G. T., Clum, A., Cramer, G. R. (2016). Phased diploid genome assembly with single-molecule real-time sequencing. *Nature Methods*, 13(12), 1050–1054.
- Clark, S. C., Egan, R., Frazier, P. I., & Wang, Z. (2013). ALE: A generic assembly likelihood evaluation framework for assessing the accuracy of genome and metagenome assemblies. *Bioinformatics*, 29(4), 435–443.

- Conesa, A., Götz, S., García-Gómez, J. M., Terol, J., Talón, M., & Robles, M. (2005). Blast2GO: A universal tool for annotation, visualization and analysis in functional genomics research. *Bioinformatics (Oxford, England)*, *21*(18), 3674–3676.
- Delcher, A. L., Phillippy, A., Carlton, J., & Salzberg, S. L. (2002). Fast algorithms for large-scale genome alignment and comparison. *Nucleic Acids Research*, *30*(11), 2478–2483.
- Dohm, J. C., Lottaz, C., Borodina, T., & Himmelbauer, H. (2008). Substantial biases in ultra-short read data sets from high-throughput DNA sequencing. *Nucleic Acids Research*, *36*(16), e105.
- English, A. C., Richards, S., Han, Y., Wang, M., Vee, V., Qu, J., Gibbs, R. A. (2012). Mind the gap: Upgrading genomes with Pacific Biosciences RS long-read sequencing technology. *PLoS One*, *7*(11), e47768.
- Eklom, R., & Wolf, J. B. (2014). A field guide to whole-genome sequencing, assembly and annotation. *Evolutionary Applications*, *7*(9), 1026–1042.
- Fondi, M., Orlandini, V., Corti, G., Severgnini, M., Galardini, M., Pietrelli, A., Fani, R. (2014). Enly: Improving draft genomes through reads recycling. *Journal of Genomics*, *2*, 89.
- Giang, V. N. L., Waminal, N. E., Park, H. S., Kim, N. H., Jang, W., Lee, J., & Yang, T. J. (2020). Comprehensive comparative analysis of chloroplast genomes from seven *Panax* species and development of an authentication system based on species-unique single nucleotide polymorphism markers. *Journal of Ginseng Research*, *44*(1), 135–144.
- Gnerre, S., MacCallum, I., Przybylski, D., Ribeiro, F. J., Burton, J. N., Walker, B. J., Berlin, A. M. (2011). High-quality draft assemblies of mammalian genomes from massively parallel sequence data. *Proceedings of the National Academy of Sciences of the United States of America*, *108*(4), 1513–1518.
- Grabherr, M. G., Haas, B. J., Yassour, M., Levin, J. Z., Thompson, D. A., Amit, I., Chen, Z. (2011). Full-length transcriptome assembly from RNA-Seq data without a reference genome. *Nature Biotechnology*, *29*(7), 644–652.
- Guizelini, D., Raittz, R. T., Cruz, L. M., Souza, E. M., Steffens, M. B., & Pedrosa, F. O. (2016). GFinisher: A new strategy to refine and finish bacterial genome assemblies. *Scientific Reports*, *6*, 34963.
- Guo, F., Wang, D., & Wang, L. (2018). Progressive approach for SNP calling and haplotype assembly using single molecular sequencing data. *Bioinformatics (Oxford, England)*, *34*(12), 2012–2018.
- Gurevich, A., Saveliev, V., Vyahhi, N., & Tesler, G. (2013). QUAST: Quality assessment tool for genome assemblies. *Bioinformatics (Oxford, England)*, *29*(8), 1072–1075.
- Healy, J. (2010). Fuzzy k-mers and their application to comparative genome. *Genome Biology*, *11*, 207.
- Hernandez, D., François, P., Farinelli, L., Østerås, M., & Schrenzel, J. (2008). De novo bacterial genome sequencing: Millions of very short reads assembled on a desktop computer. *Genome Research*, *18*(5), 802–809.
- Heydari, M., Miclotte, G., Van de Peer, Y., & Fostier, J. (2019). Illumina error correction near highly repetitive DNA regions improves de novo genome assembly. *BMC Bioinformatics*, *20*(1), 298.
- Huang, X., & Han, B. (2014). Natural variations and genome-wide association studies in crop plants. *Annual Review of Plant Biology*, *65*, 531–551.
- Jin, J. J., Yu, W. B., Yang, J. B., Song, Y., Depamphilis, C. W., Yi, T. S., & Li, D. Z. (2020). GetOrganelle: A fast and versatile toolkit for accurate de novo assembly of organelle genomes. *Genome Biology*, *21*(1), 1–31.
- Kamal, M. S., Parvin, S., Ashour, A. S., Shi, F., & Dey, N. (2017). De-Bruijn graph with MapReduce framework towards metagenomic data classification. *International Journal of Information Technology*, *9*(1), 59–75.
- Kanehisa, M., & Goto, S. (2000). KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Research*, *28*(1), 27–30.
- Kasibhatla, S. M., Waman, V. P., Kale, M. M., & Kulkarni-Kale, U. (2016). Analysis of next-generation sequencing data in virology—Opportunities and challenges. In *Next generation sequencing—Advances, applications and challenges*.
- Kircher, M. (2012). *Analysis of high-throughput ancient DNA sequencing data*. *Ancient DNA* (pp. 197–228). Humana Press.
- Koren, S., Schatz, M. C., Walenz, B. P., Martin, J., Howard, J. T., Ganapathy, G., Phillippy, A. M. (2012). Hybrid error correction and de novo assembly of single-molecule sequencing reads. *Nature Biotechnology*, *30*(7), 693–700.
- Kurtz, S., Phillippy, A., Delcher, A. L., Smoot, M., Shumway, M., Antonescu, C., & Salzberg, S. L. (2004). Versatile and open software for comparing large genomes. *Genome Biology*, *5*(2), R12.
- Kusmirek, W., & Nowak, R. (2018). De novo assembly of bacterial genomes with repetitive DNA regions by dnaasmapplication. *BMC Bioinformatics*, *19*(1), 273.
- Lander, E. S., & Waterman, M. S. (1988). Genomic mapping by fingerprinting random clones: A mathematical analysis. *Genomics*, *2*(3), 231–239.
- Langmead, B., Trapnell, C., Pop, M., & Salzberg, S. L. (2009). Bowtie: An ultrafast memory-efficient short read aligner. *Genome Biology*, *10*(3), R25.
- Lee, H., & Kingsford, C. (2018). Kourami: Graph-guided assembly for novel human leukocyte antigen allele discovery. *Genome Biology*, *19*(1), 1–16.
- Li, H. (2016). Minimap and miniasm: Fast mapping and de novo assembly for noisy long sequences. *Bioinformatics (Oxford, England)*, *32*(14), 2103–2110.
- Li, H., & Durbin, R. (2009). Fast and accurate short read alignment with Burrows–Wheeler transform. *Bioinformatics (Oxford, England)*, *25*(14), 1754–1760.
- Lin, Y., Li, J., Shen, H., Zhang, L., Papisian, C. J., & Deng, H. W. (2011). Comparative studies of de novo assembly tools for next-generation sequencing technologies. *Bioinformatics (Oxford, England)*, *27*(15), 2031–2037.
- Liu, C. J., Wu, Y., LeVan, K. J., & SoftGenetics LLC. (2012). DNA sequence assembly methods of short reads. United States Patent 8,271,206.
- Li, R., Zhu, H., Ruan, J., Qian, W., Fang, X., Shi, Z., Li, S. (2010). De novo assembly of human genomes with massively parallel short read sequencing. *Genome Research*, *20*(2), 265–272.

- Li, Z., Chen, Y., Mu, D., Yuan, J., Shi, Y., Zhang, H., Yang, B. (2012). Comparison of the two major classes of assembly algorithms: Overlap–layout–consensus and de-bruijn-graph. *Briefings in Functional Genomics*, *11*(1), 25–37.
- Lu, C. L., Chen, K. T., Huang, S. Y., & Chiu, H. T. (2014). CAR: Contig assembly of prokaryotic draft genomes using rearrangements. *BMC Bioinformatics*, *15*(1), 381.
- Lu, H., Giordano, F., & Ning, Z. (2016). Oxford NanoporeMinION sequencing and genome assembly. *Genomics, Proteomics & Bioinformatics (Oxford, England)*, *14*(5), 265–279.
- Luo, R., Liu, B., Xie, Y., Li, Z., Huang, W., Yuan, J., Tang, J. (2012). SOAPdenovo2: An empirically improved memory-efficient short-read de novo assembler. *Gigascience*, *1*(1), 2047.
- Makinen, V., Salmela, L., & Ylisen, J. (2012). Normalized N50 assembly metric using gap-restricted co-linear chaining. *BMC Bioinformatics*, *13*(1), 255.
- Mardis, E. R. (2017). DNA sequencing technologies: 2006–2016. *Nature Protocols*, *12*(2), 213.
- Marla, S. S., Mishra, P., Maurya, R., Singh, M., Wankhede, D. P., Kumar, A., Kumar, R. (2020). Refinement of draft genome assemblies of pigeonpea (*Cajanus cajan*). *Frontiers in Genetics*, *11*, 607432.
- Martin, M. (2011). Cutadapt removes adapter sequences from high-throughput sequencing reads. *EMBnet Journal*, *17*(1), 10–12.
- Maurya, R., Singh, Y., Sinha, M., Singh, K., Mishra, P., Singh, S. K., Verma, P. K. (2020). Transcript profiling reveals potential regulators for oxidative stress response of a necrotrophic chickpea pathogen *Ascochyta blight*. *3 Biotech*, *10*(3), 1–14.
- McGrath, A. (2007). Genome sequencing and assembly. *Perspectives in Bioanalysis*, *2*, 327–355.
- Miller, J. R., Koren, S., & Sutton, G. (2010). Assembly algorithms for next-generation sequencing data. *Genomics*, *95*(6), 315–327.
- Mishra, P., Maurya, R., Gupta, V. K., Ramteke, P. W., Marla, S. S., & Kumar, A. (2019). Comparative genomic analysis of monosporial and monoteliosporic cultures for unraveling the complexity of molecular pathogenesis of *Tilletia indica* pathogen of wheat. *Scientific Reports*, *9*, 8185.
- Mullikin, J. C., & Ning, Z. (2003). The phusion assembler. *Genome Research*, *13*(1), 81–90.
- Myers, E. W., Sutton, G. G., Delcher, A. L., Dew, I. M., Fasulo, D. P., Flanigan, M. J., Anson, E. L. (2000). A whole-genome assembly of *Drosophila*. *Science*, *287*(5461), 2196–2204.
- Nadalin, F., Vezzi, F., & Policriti, A. (2012). GapFiller: A de novo assembly approach to fill the gap within paired reads. *BMC Bioinformatics*, *13*(S14), S8.
- Ng, P. C., & Kirkness, E. F. (2010). *Whole genome sequencing. Genetic variation* (pp. 215–226). Totowa, NJ: Humana Press.
- Novak, P., Neumann, P., Pech, J., Steinhaisl, J., & Macas, J. (2013). RepeatExplorer: A Galaxy-based web server for genome-wide characterization of eukaryotic repetitive elements from next-generation sequence reads. *Bioinformatics (Oxford, England)*, *29*(6), 792–793.
- Panova, M., Aronsson, H., Cameron, R. A., Dahl, P., Godhe, A., Lind, U., Blomberg, A. (2016). *DNA extraction protocols for whole-genome sequencing in marine organisms. Marine genomics* (pp. 13–44). New York, NY: Humana Press.
- Peng, Y., Lai, Z., Lane, T., Nageswara-Rao, M., Okada, M., Jasieniuk, M., Stewart, C. N. (2014). De novo genome assembly of the economically important weed horseweed using integrated data from multiple sequencing platforms. *Plant Physiology*, *166*(3), 1241–1254.
- Peterlongo, P., & Chikhi, R. (2012). Mapsembler, targeted and micro assembly of large NGS datasets on a desktop computer. *BMC Bioinformatics*, *13*(1), 1–14.
- Pevzner, P. A., Tang, H., & Waterman, M. S. (2001). An Eulerian path approach to DNA fragment assembly. *Proceedings of the National Academy of Sciences of the United States of America*, *98*(17), 9748–9753.
- Pirola, Y., Zaccaria, S., Dondi, R., Klau, G. W., Pisanti, N., & Bonizzoni, P. (2016). HapCol: Accurate and memory-efficient haplotype assembly from long reads. *Bioinformatics (Oxford, England)*, *32*(11), 1610–1617.
- Piro, V. C., Faoro, H., Weiss, V. A., Steffens, M. B., Pedrosa, F. O., Souza, E. M., & Raittz, R. T. (2014). FGAP: An automated gap closing tool. *BMC Research Notes*, *7*(1), 371.
- Pop, M. (2009). Genome assembly reborn: Recent computational challenges. *Briefings in Bioinformatics*, *10*(4), 354–366.
- Salk, J. J., Schmitt, M. W., & Loeb, L. A. (2018). Enhancing the accuracy of next-generation sequencing for detecting rare and subclonal mutations. *Nature Reviews Genetics*, *19*(5), 269.
- Salzberg, S. L., & Yorke, J. A. (2005). Beware of mis-assembled genomes. *Bioinformatics (Oxford, England)*, *21*(24), 4320–4321.
- Schwarze, K., Buchanan, J., Taylor, J. C., & Wordsworth, S. (2018). Are whole-exome and whole-genome sequencing approaches cost-effective? A systematic review of the literature. *Genetics in Medicine*, *20*(10), 1122–1130.
- Shendure, J., Balasubramanian, S., Church, G. M., Gilbert, W., Rogers, J., Schloss, J. A., & Waterston, R. H. (2017). DNA sequencing at 40: Past, present and future. *Nature*, *550*(7676), 345–353.
- Silver, D. H., Ben-Elazar, S., Bogoslavsky, A., & Yanai, I. (2013). ELOPER: Elongation of paired-end reads as a pre-processing tool for improved de novo genome assembly. *Bioinformatics (Oxford, England)*, *29*(11), 1455–1457.
- Simpson, J. T., & Durbin, R. (2012). Efficient de novo assembly of large genomes using compressed data structures. *Genome Research*, *22*(3), 549–556.
- Simpson, J. T., Wong, K., Jackman, S. D., Schein, J. E., Jones, S. J., & Birol, I. (2009). ABySS: A parallel assembler for short read sequence data. *Genome Research*, *19*(6), 1117–1123.
- Stanke, M., Keller, O., Gunduz, I., Hayes, A., Waack, S., & Morgenstern, B. (2006). AUGUSTUS: Ab initio prediction of alternative transcripts. *Nucleic Acids Research*, *34*(2), W435–W439.
- Stein, L. (2001). Genome annotation: From sequence to biology. *Nature Reviews Genetics*, *2*(7), 493–503.
- Su, X., Pan, W., Song, B., Xu, J., & Ning, K. (2014). Parallel-META 2.0: Enhanced metagenomic data analysis with functional annotation, high performance computing and advanced visualization. *PLoS One*, *9*(3), e89323.

- Swain, M. T., Tsai, I. J., Assefa, S. A., Newbold, C., Berriman, M., & Otto, T. D. (2012). A post-assembly genome-improvement toolkit (PAGIT) to obtain annotated genomes from contigs. *Nature Protocols*, *7*(7), 1260–1284.
- Tiwari, A., Avashthi, H., Jha, R., Srivastava, A., Garg, V. K., Ramteke, P. W., & Kumar, A. (2016). Insights using the molecular model of Lipoxygenase from Finger millet (*Eleusinecoracana* (L.)). *Bioinformation*, *12*(3), 156.
- Torresen, O. K., Star, B., Mier, P., Andrade-Navarro, M. A., Bateman, A., Jarnot, P., Anisimova, M. (2019). Tandem repeats lead to sequence assembly errors and impose multi-level challenges for genome and protein databases. *Nucleic Acids Research*, *47*(21), 10994–11006.
- Tritt, A., Eisen, J. A., Facciotti, M. T., & Darling, A. E. (2012). An integrated pipeline for de novo assembly of microbial genomes. *PLoS One*, *7*(9), e42304.
- Tsai, I. J., Otto, T. D., & Berriman, M. (2010). Improving draft assemblies by iterative mapping and assembly of short reads to eliminate gaps. *Genome Biology*, *11*(4), R41.
- Vicedomini, R., Vezzi, F., Scalabrin, S., Arvestad, L., & Policriti, A. (2013). GAM-NGS: Genomic assemblies merger for next generation sequencing. *BMC Bioinformatics*, *14*(S7), S6.
- Walve, R., Rastas, P., & Salmela, L. (2019). Kermit: Linkage map guided long read assembly. *Algorithms for Molecular Biology*, *14*(1), 8.
- Wheeler, D. A., Srinivasan, M., Egholm, M., Shen, Y., Chen, L., McGuire, A., Gomes, X. (2008). The complete genome of an individual by massively parallel DNA sequencing. *Nature*, *452*(7189), 872–876.
- Wick, R. R., Judd, L. M., Gorrie, C. L., & Holt, K. E. (2017). Unicycler: Resolving bacterial genome assemblies from short and long sequencing reads. *PLoS Computational Biology*, *13*(6), e1005595.
- Xiao, C. L., Chen, Y., Xie, S. Q., Chen, K. N., Wang, Y., Han, Y., Xie, Z. (2017). MECAT: Fast mapping, error correction, and de novo assembly for single-molecule sequencing reads. *Nature Methods*, *14*(11), 1072.
- Xu, G. C., Xu, T. J., Zhu, R., Zhang, Y., Li, S. Q., Wang, H. W., & Li, J. T. (2019). LR_Gapcloser: A tiling path-based gap closer that uses long reads to complete genome assembly. *GigaScience*, *8*(1), giy157.
- Yandell, M., & Ence, D. (2012). A beginner's guide to eukaryotic genome annotation. *Nature Reviews Genetics*, *13*(5), 329–342.
- Zerbino, D. R., & Birney, E. (2008). Velvet: Algorithms for de novo short read assembly using de Bruijn graphs. *Genome Research*, *18*(5), 821–829.
- Zimin, A. V., Marçais, G., Puiu, D., Roberts, M., Salzberg, S. L., & Yorke, J. A. (2013). The MaSuRCA genome assembler. *Bioinformatics (Oxford, England)*, *29*(21), 2669–2677.