Hindawi Publishing Corporation
BioMed Research International
Volume 2016, Article ID 6329217, 10 pages
http://dx.doi.org/10.1155/2016/6329217

Review Article

The A, C, G, and T of Genome Assembly

Bilal Wajid,[1] Muhammad U. Sohail,[2] Ali R. Ekti,[3] and Erchin Serpedin[4]

[1]*Department of EE, University of Engineering & Technology, Lahore, Punjab 54890, Pakistan*
[2]*Department of Physiology, Government College University, Faisalabad, Punjab 38000, Pakistan*
[3]*Department of ECE, Gannon University, Erie, PA 16501, USA*
[4]*Department of ECE, Texas A&M University, College Station, TX 77840, USA*

Correspondence should be addressed to Bilal Wajid; bilalwajidabbas@hotmail.com

Received 28 September 2015; Accepted 22 December 2015

Academic Editor: Sankar Subramanian

Genome assembly in its two decades of history has produced significant research, in terms of both biotechnology and computational biology. This contribution delineates sequencing platforms and their characteristics, examines key steps involved in filtering and processing raw data, explains assembly frameworks, and discusses quality statistics for the assessment of the assembled sequence. Furthermore, the paper explores recent Ubuntu-based software environments oriented towards genome assembly as well as some avenues for future research.

1. Introduction

Genome assembly involves taking smaller fragments, called "reads," and assembling them together to form a cohesive unit, called the "sequence." However, simply assembling all the reads into one contiguous sequence, a "contig," is not enough. One has to ensure that the assembled sequence does indeed resemble what is truly present in the cell. Some common hurdles are low coverage areas, false positive read-read alignments, false negative alignments, poor sequence quality, polymorphism, and repeated regions of the genome. An even more fundamental concern lies in the difficulty of determining which of the two strands was finally reported in the sequencing procedure. Moreover, as a number of research domains draw suitable conclusions from the sequence itself, a sequence that has not been reported accurately may potentially affect subsequent analyses [1].

Sanger's deoxydinucleotide sequencing with large and accurate reads opened the door to whole-genome sequencing and deciphered the first human genome in 2001 [2, 3]. Sanger's approach is still commercially available with improved capillary electrophoresis, enhanced speed and accuracy, and longer read lengths. NIH's $1,000 genome project led researchers to develop efficient, economical, and high-throughput sequencing platforms introducing a new paradigm called next-generation sequencing (NGS). For instance, Roche's 454 GS, Illumina's MiSeq and HiSeq, ABI's SOLiD, and Life Technologies' Ion Torrent and Proton Torrent platforms all sequence the same genome at a fraction of the time and cost of the first-generation sequencing methods [4].

NGS platforms now produce terabytes of data thereby challenging traditional software tools and hardware architectures which were not designed to process such large amounts of data. This triggered a need to develop algorithms and statistical tools with improved memory management and time complexity in parallel to the development of NGS platforms.

This contribution is intended to act as an introductory note to scientists and researchers working in the area of genome assembly. Section 2 provides an overview of NGS platforms. Section 3 discusses raw data, Sequencing Read Archive, and FASTA and FASTQ file formats. It provides particulars on filtering and correcting raw data. Additionally, the second section enforces the need to report accurate results. Section 4 supplies necessary answers addressing the draft assembly process. Section 5 reviews common metrics employed to evaluate the assembly and Section 6 highlights recent software environments oriented towards NGS. Finally, Section 7 projects considerations on possible future research trends.

2. Overview of Next-Generation Sequencing Platforms

Among NGS platforms, Roche's 454 sequencing is based on Nyren's pyrosequencing approach [5]. Roche's approach, referred to as "sequencing by synthesis" (SS), takes one DNA strand as a template and then uses it to synthesize the sequence of its complementary strand. Roche's SS uses four polymerase enzymes to extend several DNA strands in parallel. Whenever a nucleotide attaches itself onto template DNA, a pyrophosphate molecule is produced which emits light when triggered [6]. The bioluminescence produced by these bases helps in recognizing the bases and, therefore, the sequence. Some characteristics of Roche sequencing include its automated procedures and high speed, while some drawbacks are lower read accuracy for homopolymer segments of identical bases and relatively high operating costs [7].

Illumina, another NGS company, differs from Sanger in several features. Sanger's approach uses dideoxynucleotide for irreversible termination of primer extension, whereas Illumina employs reversible terminators for primer extension of the complementary strand. Illumina's 3-O-azidomethyl reversible terminators are tagged with four different colored fluorophores to distinguish between the four nucleotides. Therefore, using these reversible terminators aids in observing the identity of the nucleotides as they attach onto the DNA fragment because the fluorophores are detected by highly sensitive CCD cameras [8]. Illumina's method significantly reduces the duration of sequencing and assumes a \$1000 price tag for 30× human genome. Illumina's sequencing scheme shows some benefits over Roche's pyrosequencing; however, its characteristic short read lengths (<300 bp) present challenges when resolving short sequence repeats.

In addition to Roche and Illumina, Applied Biosystems' SOLiD sequencer is another key player among genome sequencers. SOLiD uses the principle of "sequencing by ligation" (SL). SL differs from Illumina in its method for ligation of octamer oligonucleotides. SL uses dibase fluorescent labeled octaoligonucleotide adaptors which link the template DNA and are bound with 1 μm magnetic beads [9]. At each step, SOLiD's technique encrypts two bases simultaneously and every nucleotide is cross-examined twice: first as the right nucleotide of a pair and then as the left one. This approach reduces homopolymeric sequencing errors. However, similar to Illumina, SOLiD generates short read length data which incur complications in the sequence assembly.

Collectively, these high-throughput sequencers have substantially reduced the cost (≤\$0.1/Mb) and duration of genome sequencing. However, additional technologies with enhanced performance have been proposed recently. The advent of nonoptic, semiconductor-based genome sequencers has shown potential. Manufacturers like Life Technologies developed Ion Proton and Ion PGM, both of which use SS amplification and hydrogen ion sensing semiconductors [10]. The sequence is obtained by sensing hydrogen ions emitted when nucleotides incorporate themselves onto template DNA, a process catalyzed by DNA polymerase. Massively parallel transistor-based integrated circuits with about two million wells allow simultaneous detection of multiple reactions. Furthermore, signal processing tools translate voltage fluctuations into base calls for successive nucleotides [10].

Another technique which has been recently proposed is the single-molecule real-time (SMRT) sequencing, introduced by HeliScope [11]. SMRT sequencing scheme is free of library preparation or amplification errors. PacBio RS II (by HeliScope) utilizes SMRT sequencing and can produce about 50,000 reads ranging from 15,000 to 40,000 bases in length in just three hours. The extended read length facilitates sequence alignment and improves precision in drafting an assembly, simply because long repetitive DNA fragments can be easily spanned. Interestingly, Roche will be phasing out its in-house 454 sequencers in 2016 in favor of PacBio's SMRT sequencers. Roche plans to maintain its participation in NGS market, not by developing its own sequencers, but rather by becoming an exclusive seller for in vitro diagnostic products based on PacBio's SMRT sequencing platform (http://www.bio-itworld.com/BioIT_Article.aspx?id=131053, accessed on Dec. 12, 2015). Together with nonoptic semiconductor nanopore technology, SMRT sequencers are referred to as "third-generation-sequencers" [12–14]. Overall, the above-mentioned high-throughput sequencers have substantially reduced the duration and cost of sequencing (\$0.1/Mb).

Companies are investing significant resources to upgrade existing technologies and introduce newer machines. It is hoped that many third-generation-sequencers are expected to surface, coupling SMRT sequencing with principles of electrothermodynamics, quantum physics, and nanopore technology [13–15]. Existing platforms are currently designed to cater for de novo synthesis, wholegenome/whole-exome and transcriptome synthesis, targeted resequencing, RNA profile ChIP-Seq, mutation detection, and metagenomics. Platforms are usually accompanied by bioinformatics tools. Tables 1, 2, and 3 present some important details about current sequencers.

3. Preliminary Data Processing Steps

Software tools and applications enter the research process once the sequencers fulfill their role of generating reads. The aim of this and the next set of sections is to provide an outline of the individual steps involved in transforming raw data into the novel genome, as presented in Figure 1. The set of interconnected methods are referred to as a "pipeline." The process starts by using the data generated by one's lab or by downloading the data from the Sequencing Read Archive (SRA) [16]. Data is present in ".SRA" format and must be converted into .FASTQ file format by employing the SRA toolkit (http://www.ncbi.nlm.nih.gov/Traces/sra/). Once converted, the FASTQ format adopts a four-line representation to display the sequence and its associated quality [1]:

@ Sequence Identifier

Sequence line(s)

+ Sequence Identifier

ASCII encoding of quality values

Table 1: Comparison of current (as of Nov. 15, 2014) sequencing platforms. PCR: polymerase chain reaction, SS: sequencing by synthesis, SL: sequencing by ligation, SH: sequencing by hybridization, and SE: sequencing by expansion.

Platform	Biochemistry/ biotechnology	Amplification	Throughput	Reads per run	Read length (bp)	Seq run time	Error rate (%)	Machine cost ×1000	Cost per run	Cost per unit data
Sanger (Applied Biosystems 3730xl)	Dideoxynucleotide termination of PCR	PCR	0.06 Mb	9600	1000	2 hrs	0.1	$100	$100	$8,000–$10,000/Gb
454 GS+	Bioluminescence on nucleotide incorporation	Emulsion PCR	~70 Mb	70 k~100 k	~700	18 hours	<1.0	$125	$1,000	$28.50/Gb
454 GS FLX+	Bioluminescence on nucleotide incorporation	Emulsion PCR	700 Mb	1 M	~1000	23 hours	<1.0	$500	$6,000	$8.50/Gb
MiSeq	Cleavage of 3′-O-azidomethyl reversible terminator and fluorescent tag on nucleotide incorporation	SS	15 Gb	25 M	2W300	5~ 55 hrs	0.1	$125	$1.4K	$93/Gb
HiSeq X Ten	Cleavage of 3′-O-azidomethyl reversible terminator and fluorescent tag on nucleotide incorporation	SS	1000 Gb	4000 M	2W125	7 hrs~ 6 d	0.1	$1,000	$12K	$7/Gb
NextSeq 500	Cleavage of 3′-O-azidomethyl reversible terminator and fluorescent tag on nucleotide incorporation	SS	129 Gb	400 M	2W150	26~ 29 hrs	0.1	$250	$4K	$33/Gb
SOLiD 5500xl	Ligation of octamer oligonucleotide and cleavage of fluorescent tag	SL	180 Gb	2.8 B	2W60	150 hrs	0.01	$595	$10K	$9/Gb
Ion Proton I	Proton sensing by pH change	SS	10 Gb	40~80 M	200	2~4 hrs	1.0	$149	$1K	$100/Gb
Ion PGM 318	Proton sensing by pH change	SS	2 Gb	5 M	400	7.3 hrs	1.0	$52	$750	$350/Gb
Polonator G.007	Cleavage of 3′-ONH2 reversible terminator and fluorescent tag on nucleotide incorporation	SL	10 Gb	—	26	—	N.A	N.A	N.A	N.A
Helicos HeliScope	Single-molecule real-time sequencing	SS	35 Gb	20 M	35	8 hrs	0.5	$1,000	$10K	$330/Gb
PacBio RS II	Single-molecule real-time sequencing	SS	1 Gb	50,000	15,000 bp	3 hrs	15	$700	$400	~$1000/Gb

Table 2: The table enlists the strong points and challenges pertaining to some of the sequencing platforms.

Platform	Positive points	Challenges
Sanger (Applied Biosystems 3730xl)	Long read length; good for individual gene analysis	Slow; expensive; poor quality due to primer dimer
454 GS+	Long read length; fast; low cost for small studies	High error rate for homopolymer read; low throughput; will be phased out in 2016
454 GS FLX+	Long read length	High error rate homopolymer read; low throughput; large capital cost; will be phased out in 2016
MiSeq	High throughput; ideal for small genome project	Short read length
HiSeq X Ten	High throughput; ideal for whole-genome project	Short read length
NextSeq 500	High throughput; ideal for small to large scale project	Short read length
SOLiD 5500xl	High throughput	Short read length; poor output data distribution and arduous data analysis
Ion Proton I	Ideal for small project; shorter run time; leading future technology	Higher error rate; larger cost per Mb
Ion PGM 318	Low capital investment and running cost; shorter run time	Higher error rate; larger cost per Mb
Polonator G.007	Cost-effective; open resource	Obsolete
Helicos HeliScope	Single-molecule sequencing; simple sample preparation and data analysis	Short read length; obsolete
PacBio RS II	Single-molecule real-time sequencing; longest available read length	High error rate

Table 3: Recent sequencing platforms: these platforms are relatively new and to date (Nov. 15, 2014) there is not enough information to incorporate them into Table 1.

Platform	Company	Biotechnology	Resource
GENIUS	GenapSys	Proton sensing by pH and temperature change	http://genapsys.com/
NanoTag sequencer	Genia	Electric current change produced by nanotag released from incorporation of nucleotide	http://geniachip.com/
GnuBIO platform	GnuBIO system	Oligo hexamers hybridization in microfluidics	http://gnubio.com/
*	Lasergene	3′-OH unblocked reversible terminator	http://lasergen.com/
*	Nabsys	Hexamer oligonucleotides hybridization mapping through nanopore arrays	http://nabsys.com/
MinION and GridION	Oxford Nanopore Technologies	Strand DNA or exonuclease cleaved nucleotides pass through nanopores change electric current flow rate	https://nanoporetech.com/
*	Strato Genomics Technology	Conversion of DNA into Xpandomer	http://stratosgenomics.com/

*Lasergene, Nabsys, and Strato Genomics are working on newer platforms.

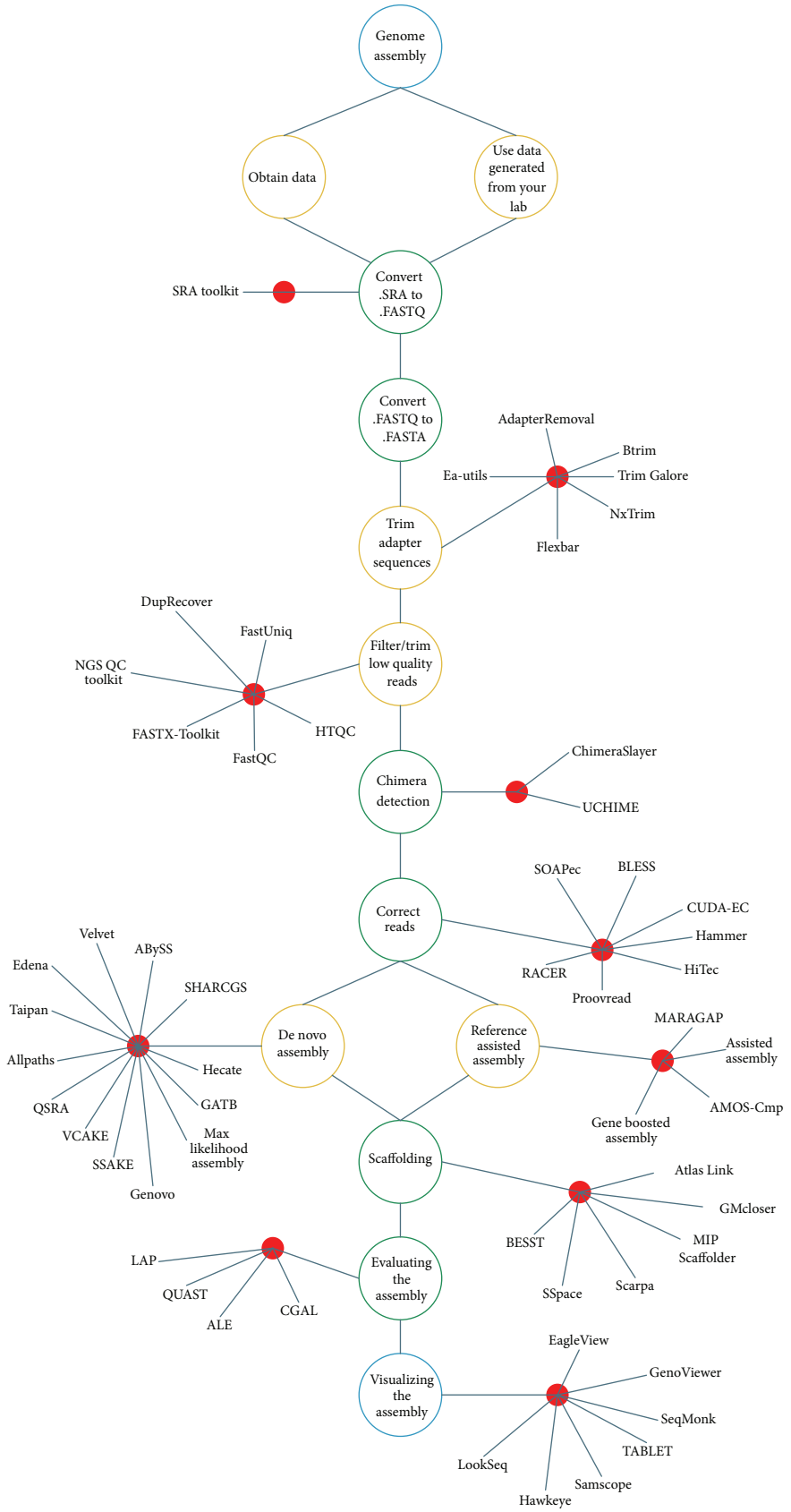

Figure 1: Flow chart for DNA assembly pipeline. Some commonly used tools are mentioned next to each step [36]. Please refer to [19, 35, 37–88] for details on the above-mentioned tools.

ASCII characters utilized in the last line of the above-mentioned SRA format symbolize quality values (Q-values). Q-values are log-probabilities illustrating the quality of each base call. For example, for Sanger the formula is

$$Q_{\text{PHRED}} = -10 \times \log_{10}(P_e), \quad (1)$$

where P_e is the probability of determining a base incorrectly [17, 18]. For ASCII encoded quality values the following characters depict an increasing order of quality:

!"#$%&'()*+,-./0123456789:;<=>?@*ABCDEFG*
HIJKLMNOPQRSTUVWXYZ[\]^_`*abcdefghijklmno*
pqrstuvwxyz{|}~

Similar to FASTQ, FASTA format seems like an abridged version of FASTQ file format. It maintains a two-line arrangement to display the sequence and contains no mention of its quality:

@ Sequence Identifier
sequence line(s)

Once reads are received in their correct format, one must trim adapter sequences, filter, or trim low quality ends and collapse identical reads. A naive approach is to remove all reads that contain the flag "N." An improved method retains all reads that have an overall quality $P_{\text{qual}} > q$, where q is a user-defined parameter [19–23]. A more enhanced approach consists in matching reads against known ribosomal and heterochromatin DNA and removing them should they match [24]. Nevertheless, since a significant portion of raw data contains errors one must correct them.

4. Assembly Process

The primary aim of the assembly process is to connect all reads together, one after another, to form a single contiguous sequence. Interestingly, due to the inherent nature of the problem, graph theory, especially de Bruijn graph, models very well such a process [25]. In graphical models individual nodes symbolize reads whereas edges between the nodes emphasize "overlaps" between reads. Once the overlap between all reads is established, the task at hand is to generate a "layout" by searching for a single path from beginning, that is, the root of the graph structure, to the end, the leaf of the graph structure, as illustrated in Figures 2 and 3. As such, generating a layout is very challenging, because not one but multiple disjointed graphs are realized, each depicting a contig. In addition, each graph has many loops portraying repeat regions as well as multiple branches, both long and short. All these hazards need to be resolved. Branches that are small may be discarded, while longer branches compete with one another to serve as potential representatives for each contig. Loops portray repeat regions, so one must decide how many times the repeats should be placed within the final assembly. Nevertheless, assemblers do spend a significant amount of time resolving potential hazards, in multiple ways. The output is a collection of contigs that need to be ordered, appended, and elongated, a process called "scaffolding" [25–28].

Figure 2: De novo assembly: reads that *overlap* each other are shown to align at appropriate places with respect to one another, thereby generating the *layout*. The layout, in turn, constructs a *consensus* sequence, simply by basing itself on the majority base call. The above-mentioned framework is called "*Overlap-Layout-Consensus*."

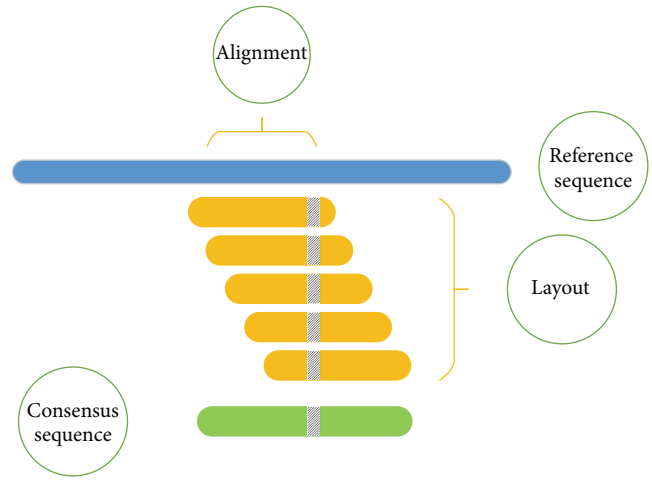

Figure 3: *Reference assisted assembly*: reads *align* relative to a reference sequence setting up the *layout*. The layout, in turn, constructs a *consensus* sequence, simply by basing itself on the majority base call. Please note that the reads do not need to match perfectly with the reference. The example shows a shaded region where the consensus sequence differs from the reference. This working scheme is called "*Alignment-Layout-Consensus*."

5. Evaluating the Quality of an Assembly

Evaluating the quality of an assembly requires analyzing multiple metrics. These statistics measure an assembly from various standpoints. Table 4 illustrates some commonly used assembly metrics/statistics and their explanations. After evaluating the assembly it is recommended to visualize the assembly in order to obtain a pictorial view of the draft. Figure 1 presents common tools used in each part of the pipeline.

6. Linux Based Distributions

The software environments pertaining to genome assembly are many and as such need to be constantly maintained, configured, and updated. This repeated and continuous configuration consumes a good amount of time and resources. Therefore, to address these challenges, engineers and computer scientists

Table 4: Some common assembly statistics. Here an ↑ indicates higher is better while a ↓ implies less is better.

↑ / ↓	Description
↑	*N50*: quantified the length of the scaffold at which 50% of the total assembled size of the sequence is covered. *NG50*: evaluated in a way similar to N50. However, here the length of the sequence is either known or predicted [1, 29]. *NA50* and *NGA50*: these metrics deal with aligned blocks rather than contigs [35]. *Continuity*: similar to N50, NA50, NG50, and NGA50 there are other metrics like *N75, NA75, NG75, NGA75, N90, NA90, NG90*, and *NGA90*. *Number of Genes*: an assembly which exhibits more highly conserved core Eukaryotic genes is considered better [29]. *Accuracy*: if an assembly reports at least 90% of its bases with a minimum of 5× coverage, it is considered accurate. *Choppiness*: the average contig length should be greater than a certain threshold. Otherwise, the assembly needs to be redrafted. *Validity*: the fraction of assembly that can be confirmed by a reference sequence [29]. *Completeness*: an assembly is considered complete if the scaffolds cover more than 90% of the actual genome. *Length of the Longest Scaffold*: typically the greater the length, the better the assembly. Similar is the case of the *shortest scaffold*. *Number of scaffolds* > *X*, where *X* is a user-defined length. Similarly, % *age of scaffolds* > *X*. *Total Length of the Scaffolds* and *Total Scaffold Length as Percentage of Estimated Genome Size*: the closer it is to 100%, the better it is. *Percentage of Contigs Scaffolded*: percentage of contigs that were connected with one another during the scaffolding process [1].
↓	*Number of Gaps in the Assembly*: by aligning paired-read data onto scaffolds one may determine scaffolding errors [1]. *Number of Scaffolds*: an assembly which has a smaller number of scaffolds would be assumed to be better. For example, the optimum assembly would be one continuous sequence depicting the true sequence. *LG50 Scaffold Count*: number of scaffolds counted in reaching NG50 threshold. Similar would be the case of *LG75* and *LG90*. *Percentage of Unscaffolded Contigs*: since contigs may remain unscaffolded.

Table 5: Comparison of different Linux distributions. Here LTS stands for Long Term Support and GUI refers to Graphical User Interface.

Operating system	Free	Base OS	Software	Open source	LTS	GUI	×86/×64	Cloud	Script files
Baari	√	Ubuntu 13.10	60+ genome assembly tools	√	√	Unity	×64	×	√
Lxtoo	√	Gentoo Linux 11	Sequence analysis, protein-protein interactions	√	√	X11 Desktop	×86/×64	×	×
Open Discovery 3	×	Fedora Sulphur 9	Molecular dynamics, docking, sequence analysis	×	√	GNOME 2.22	×86/×64	√	×
BioBrew	√	Red Hat 7.3	Appropriate for clusters	√	×	KDE, GNOME	×86	×	×
PhyLIS	√	Ubuntu 8	Phylogenetics	√	×	Unity	×86/×64	×	×
DNALinux	√	Xubuntu	DNA and protein analysis	√	×	XFCE 4.2.2	×86	√	×
Bioconductor Buntu	√	Ubuntu 12.04	Bioconductor Buntu 2.11	√	√	Unity	×86/×64	×	×
BioLinux 7	√	Ubuntu 12.04	500+ bioinformatics applications with 7 assembly tools	√	√	Unity	×64	√	×

have proposed multiple solutions built on Linux systems that include within them all the necessary software needed by the research group. Table 5 mentions a few. As for genome assembly, both Baari, an Ubuntu-derived operating system (http://people.tamu.edu/~bilalwajidabbas/Baari.html), and Genobuntu, a software package, provide about 60+ genome assembly tools (https://sourceforge.net/projects/genobuntu/). It is hoped the current set of tools will be constantly updated to suit the ever growing needs of the scientific community.

7. Considerations and Concerns

Genome Online Database (GOLD) reports that as of Dec 12, 2015, 1,136 Archaeal, 49,983 Bacterial, 4,473 Viruses, and 11,122 Eukaryotic genomes have been sequenced. There remains plenty of room for work. The $1000 genome project has reduced the cost significantly, but if personalized medication is expected to be effective and available to everyone, the cost and time duration for sequencing need to be reduced further. Processing raw data needs to be done both cheaply and at ultra-fast rates. Spending about 50 hours of processing time on a system with 20 microprocessor cores and 20 GB RAM is not uncommon (as of 2014) [29]. Imagine trying to sequence the genomes of an entire country's population. Transferring all the raw data via an Internet connection from one country to another is not feasible. Therefore, countries will have to provide for their own supercomputers, and algorithms will need to be parallelized with careful

attention to Hadoop and MapReduce frameworks [30–34]. Hadoop and MapReduce are ideal as both are designed to process "big-data" using parallel and distributed algorithms on clusters of systems [30–34]. With so many obstacles ahead, genome assembly will remain challenging for many years to come.

Key Points

(i) NIH's $1,000 genome project led researchers to develop efficient, economical, and high-throughput sequencing platforms. Examples include Roche's 454 GS, Illumina's MiSeq and HiSeq, ABI's SOLiD, and Life Technologies' Ion Torrent and Proton Torrent platforms. A brief comparison of these next-generation sequencing platforms is presented.

(ii) Data provided by these platforms is transformed into a sequence via a series of processes collectively called a "pipeline." It starts with trimming adapter sequences, filtering low quality ends, and collapsing identical reads. The final set of reads are then connected together, one after another, to form contiguous sequences, called "contigs." The collection of contigs needs to be ordered, appended, and elongated via a process called "scaffolding."

(iii) A number of software environments providing bioinformatics solutions have been provided over the years. A brief comparison of some of these is presented here.

Conflict of Interests

The authors declare no conflict of interests regarding the publication of this paper.

References

[1] B. Wajid and E. Serpedin, "Do it yourself guide to genome assembly," *Briefings in Functional Genomics*, vol. 15, no. 1, pp. 1–9, 2016.

[2] J. C. Venter, M. D. Adams, E. W. Myers et al., "The sequence of the human genome," *Science*, vol. 291, no. 5507, pp. 1304–1351, 2001.

[3] F. Sanger, S. Nicklen, A. R. Coulson et al., "DNA sequencing with chain-terminating inhibitors," *Biotechnology*, vol. 74, no. 12, pp. 5463–5467, 1992.

[4] D. A. Wheeler, M. Srinivasan, M. Egholm et al., "The complete genome of an individual by massively parallel DNA sequencing," *Nature*, vol. 452, no. 7189, pp. 872–876, 2008.

[5] A. Ahmadian, B. Gharizadeh, A. C. Gustafsson et al., "Single-nucleotide polymorphism analysis by pyrosequencing," *Analytical Biochemistry*, vol. 280, no. 1, pp. 103–110, 2000.

[6] A. Ahmadian, M. Ehn, and S. Hober, "Pyrosequencing: history, biochemistry and future," *Clinica Chimica Acta*, vol. 363, no. 1-2, pp. 83–94, 2006.

[7] O. Morozova and M. A. Marra, "Applications of next-generation sequencing technologies in functional genomics," *Genomics*, vol. 92, no. 5, pp. 255–264, 2008.

[8] D. R. Bentley, S. Balasubramanian, H. P. Swerdlow et al., "Accurate whole human genome sequencing using reversible terminator chemistry," *Nature*, vol. 456, no. 7218, pp. 53–59, 2008.

[9] J. Shendure, G. J. Porreca, N. B. Reppas et al., "Molecular biology: accurate multiplex polony sequencing of an evolved bacterial genome," *Science*, vol. 309, no. 5741, pp. 1728–1732, 2005.

[10] J. M. Rothberg, W. Hinz, T. M. Rearick et al., "An integrated semiconductor device enabling non-optical genome sequencing," *Nature*, vol. 475, no. 7356, pp. 348–352, 2011.

[11] P. Milos, "Helicos BioSciences," *Pharmacogenomics*, vol. 9, no. 4, pp. 477–480, 2008.

[12] J. Eid, A. Fehr, J. Gray et al., "Real-time DNA sequencing from single polymerase molecules," *Science*, vol. 323, no. 5910, pp. 133–138, 2009.

[13] T. P. Niedringhaus, D. Milanova, M. B. Kerby, M. P. Snyder, and A. E. Barron, "Landscape of next-generation sequencing technologies," *Analytical Chemistry*, vol. 83, no. 12, pp. 4327–4341, 2011.

[14] B. M. Venkatesan and R. Bashir, "Nanopore sensors for nucleic acid analysis," *Nature Nanotechnology*, vol. 6, no. 10, pp. 615–624, 2010.

[15] E. E. Schadt, S. Turner, and A. Kasarskis, "A window into third-generation sequencing," *Human Molecular Genetics*, vol. 19, no. 2, pp. R227–R240, 2010.

[16] R. Leinonen, H. Sugawara, and M. Shumway, "The sequence read archive," *Nucleic Acids Research*, vol. 39, no. 1, pp. D19–D21, 2011.

[17] P. J. A. Cock, C. J. Fields, N. Goto, M. L. Heuer, and P. M. Rice, "The Sanger FASTQ file format for sequences with quality scores, and the Solexa/Illumina FASTQ variants," *Nucleic Acids Research*, vol. 38, no. 6, pp. 1767–1771, 2010.

[18] B. Wajid, M. Nounou, H. Nounou et al., "Gibbs-beca: gibbs sampling and bayesian estimation for comparative assembly," in *Proceedings of the 3rd International Conference on Biomedical Engineering, Electronics and Nanotechnology (MIC-BEN '13)*, vol. 3, Mosharaka for Researches and Studies, 2013.

[19] R. K. Patel and M. Jain, "NGS QC Toolkit: a toolkit for quality control of next generation sequencing data," *PLoS ONE*, vol. 7, no. 2, Article ID e30619, 2012.

[20] B. Yuan, "Mapping next generation sequence reads," 2010.

[21] S. P. Mane, T. Modise, and B. W. Sobral, "Analysis of high-throughput sequencing data," *Methods in Molecular Biology*, vol. 678, pp. 1–11, 2011.

[22] A. Gordon and G. Hannon, "Fastx-toolkit," 2010, http://hannonlab.cshl.edu/fastx_toolkit/index.html.

[23] J. Goecks, A. Nekrutenko, J. Taylor et al., "Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences," *Genome Biology*, vol. 11, no. 8, article R86, 2010.

[24] E. W. Myers, G. G. Sutton, A. L. Delcher et al., "A whole-genome assembly of Drosophila," *Science*, vol. 287, no. 5461, pp. 2196–2204, 2000.

[25] B. Wajid and E. Serpedin, "Review of general algorithmic features for genome assemblers for next generation sequencers," *Genomics, Proteomics and Bioinformatics*, vol. 10, no. 2, pp. 58–73, 2012.

[26] J. R. Miller, S. Koren, and G. Sutton, "Assembly algorithms for next-generation sequencing data," *Genomics*, vol. 95, no. 6, pp. 315–327, 2010.

[27] S. Meader, L. W. Hillier, D. Locke, C. P. Ponting, and G. Lunter, "Genome assembly quality: assessment and improvement using

the neutral indel model," *Genome Research*, vol. 20, no. 5, pp. 675–684, 2010.

[28] C. Alkan, S. Sajjadian, and E. E. Eichler, "Limitations of next-generation genome sequence assembly," *Nature Methods*, vol. 8, no. 1, pp. 61–65, 2011.

[29] K. R. Bradnam, J. N. Fass, A. Alexandrov et al., "Assemblathon 2: evaluating de novo methods of genome assembly in three vertebrate species," *Gigascience*, vol. 2, no. 1, article 10, 2013.

[30] T. White, *Hadoop: The Definitive Guide*, O'Reilly Media, Sebastopol, Calif, USA, 2012.

[31] A. Zomaya, *Parallel Computing for Bioinformatics and Computational Biology*, John Wiley & Sons, New York, NY, USA, 2006.

[32] E. Talbi and A. Zomaya, *Grid Computing for Bioinformatics and Computational Biology*, vol. 1, John Wiley & Sons, New York, NY, USA, 2008.

[33] J. Augen, *Bioinformatics in the Post-Genomic Era: Genome, Transcriptome, Proteome, and Information-Based Medicine*, Addison-Wesley Professional, 2004.

[34] Y. Chen, *Bioinformatics Technologies*, Springer, New York, NY, USA, 2005.

[35] A. Gurevich, V. Saveliev, N. Vyahhi, and G. Tesler, "QUAST: quality assessment tool for genome assemblies," *Bioinformatics*, vol. 29, no. 8, pp. 1072–1075, 2013.

[36] V. J. Henry, A. E. Bandrowski, A. S. Pepin, B. J. Gonzalez, and A. Desfeux, "OMICtools: an informative directory for multi-omic data analysis," *Database*, vol. 2014, Article ID bau069, 2014.

[37] S. Lindgreen, "AdapterRemoval: easy cleaning of next-generation sequencing reads," *BMC Research Notes*, vol. 5, article 337, 2012.

[38] E. Aronesty, "Comparison of sequencing utility programs," *The Open Bioinformatics Journal*, vol. 7, no. 1, pp. 1–8, 2013.

[39] M. Dodt, J. T. Roehr, R. Ahmed, and C. Dieterich, "FLEXBAR-flexible barcode and adapter processing for next-generation sequencing platforms," *Biology*, vol. 1, no. 3, pp. 895–905, 2012.

[40] J. O'Connell, O. Schulz-Trieglaff, E. Carlson, M. M. Hims, N. A. Gormley, and A. J. Cox, "NxTrim: optimized trimming of Illumina mate pair reads," *Bioinformatics*, vol. 31, no. 12, pp. 2035–2037, 2015.

[41] Z. Wu, X. Wang, and X. Zhang, "Using non-uniform read distribution models to improve isoform expression inference in RNA-Seq," *Bioinformatics*, vol. 27, no. 4, pp. 502–508, 2011.

[42] Y. Kong, "Btrim: a fast, lightweight adapter and quality trimming program for next-generation sequencing technologies," *Genomics*, vol. 98, no. 2, pp. 152–153, 2011.

[43] H. Xu, X. Luo, J. Qian et al., "FastUniq: a fast de novo duplicates removal tool for paired short reads," *PLoS ONE*, vol. 7, no. 12, Article ID e52249, 2012.

[44] X. Yang, D. Liu, F. Liu et al., "HTQC: a fast quality control toolkit for Illumina sequencing data," *BMC Bioinformatics*, vol. 14, article 33, 2013.

[45] R. Schmieder and R. Edwards, "Quality control and preprocessing of metagenomic datasets," *Bioinformatics*, vol. 27, no. 6, pp. 863–864, 2011.

[46] M. P. Davis, S. V. Dongen, C. A. Goodger, N. Bartonicek, and A. J. Enright, "Kraken: a set of tools for quality control and analysis of high-throughput sequence data," *Methods*, vol. 63, no. 1, pp. 41–49, 2013.

[47] R. Bao, L. Huang, J. Andrade et al., "Review of current methods, applications, and data management for the bioinformatics analysis of whole exome sequencing," *Cancer Informatics*, vol. 13, supplement 2, pp. 67–82, 2014.

[48] R. C. Edgar, B. J. Haas, J. C. Clemente, C. Quince, and R. Knight, "UCHIME improves sensitivity and speed of chimera detection," *Bioinformatics*, vol. 27, no. 16, pp. 2194–2200, 2011.

[49] B. J. Haas, D. Gevers, A. M. Earl et al., "Chimeric 16S rRNA sequence formation and detection in Sanger and 454-pyrosequenced PCR amplicons," *Genome Research*, vol. 21, no. 3, pp. 494–504, 2011.

[50] Y. Heo, X.-L. Wu, D. Chen, J. Ma, and W.-M. Hwu, "BLESS: bloom filter-based error correction solution for high-throughput sequencing reads," *Bioinformatics*, vol. 30, no. 10, pp. 1354–1362, 2014.

[51] H. Shi, B. Schmidt, W. Liu, and W. Müller-Wittig, "A parallel algorithm for error correction in high-throughput short-read data on CUDA-enabled graphics hardware," *Journal of Computational Biology*, vol. 17, no. 4, pp. 603–615, 2010.

[52] P. Medvedev, E. Scott, B. Kakaradov, and P. Pevzner, "Error correction of high-throughput sequencing datasets with non-uniform coverage," *Bioinformatics*, vol. 27, no. 13, pp. i137–i141, 2011.

[53] L. Ilie, F. Fazayeli, and S. Ilie, "HiTEC: accurate error correction in high-throughput sequencing data," *Bioinformatics*, vol. 27, no. 3, pp. 295–302, 2011.

[54] T. Hackl, R. Hedrich, J. Schultz, and F. Förster, "Proovread: large-scale high-accuracy PacBio correction through iterative short read consensus," *Bioinformatics*, vol. 30, no. 21, pp. 3004–3011, 2014.

[55] L. Ilie and M. Molnar, "RACER: rapid and accurate correction of errors in reads," *Bioinformatics*, vol. 29, no. 19, pp. 2490–2493, 2013.

[56] X. Yang, S. P. Chockalingam, and S. Aluru, "A survey of error-correction methods for next-generation sequencing," *Briefings in Bioinformatics*, vol. 14, no. 1, pp. 56–66, 2013.

[57] R. L. Warren, G. G. Sutton, S. J. M. Jones, and R. A. Holt, "Assembling millions of short DNA sequences using SSAKE," *Bioinformatics*, vol. 23, no. 4, pp. 500–501, 2007.

[58] D. D'Agostino, A. Clematis, A. Guffanti, L. Milanesi, and I. Merelli, "A CUDA-based implementation of the SSAKE genomics application," in *Proceedings of the 20th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP '12)*, pp. 612–616, IEEE, Garching, Germany, February 2012.

[59] W. R. Jeck, J. A. Reinhardt, D. A. Baltrus et al., "Extending assembly of short DNA sequences to handle error," *Bioinformatics*, vol. 23, no. 21, pp. 2942–2944, 2007.

[60] D. W. Bryant Jr., W.-K. Wong, and T. C. Mockler, "QSRA—a quality-value guided *de novo* short read assembler," *BMC Bioinformatics*, vol. 10, article 69, 2009.

[61] J. Butler, I. MacCallum, M. Kleber et al., "ALLPATHS: de novo assembly of whole-genome shotgun microreads," *Genome Research*, vol. 18, no. 5, pp. 810–820, 2008.

[62] T. Namiki, T. Hachiya, H. Tanaka, and Y. Sakakibara, "MetaVelvet: an extension of Velvet assembler to *de novo* metagenome assembly from short sequence reads," *Nucleic Acids Research*, vol. 40, no. 20, article e155, 2012.

[63] D. R. Zerbino and E. Birney, "Velvet: algorithms for de novo short read assembly using de Bruijn graphs," *Genome Research*, vol. 18, no. 5, pp. 821–829, 2008.

[64] J. C. Dohm, C. Lottaz, T. Borodina, and H. Himmelbauer, "SHARCGS, a fast and highly accurate short-read assembly algorithm for de novo genomic sequencing," *Genome Research*, vol. 17, no. 11, pp. 1697–1706, 2007.

[65] J. T. Simpson, K. Wong, S. D. Jackman, J. E. Schein, S. J. M. Jones, and I. Birol, "ABySS: a parallel assembler for short read sequence data," *Genome Research*, vol. 19, no. 6, pp. 1117–1123, 2009.

[66] E. Drezen, G. Rizk, R. Chikhi et al., "GATB: genome assembly & analysis tool box," *Bioinformatics*, vol. 30, no. 20, pp. 2959–2961, 2014.

[67] D. Hernandez, P. François, L. Farinelli, M. Østerås, and J. Schrenzel, "De novo bacterial genome sequencing: millions of very short reads assembled on a desktop computer," *Genome Research*, vol. 18, no. 5, pp. 802–809, 2008.

[68] S. Gnerre, E. S. Lander, K. Lindblad-Toh, and D. B. Jaffe, "Assisted assembly: how to improve a de novo genome assembly by using related species," *Genome Biology*, vol. 10, no. 8, article R88, 2009.

[69] S. L. Salzberg, D. D. Sommer, D. Puiu, and V. T. Lee, "Gene-boosted assembly of a novel bacterial genome from very short reads," *PLoS Computational Biology*, vol. 4, no. 9, Article ID e1000186, 2008.

[70] M. Pop, A. Phillippy, A. L. Delcher, and S. L. Salzberg, "Comparative genome assembly," *Briefings in Bioinformatics*, vol. 5, no. 3, pp. 237–248, 2004.

[71] B. Wajid, A. R. Ekti, A. Noor et al., "Supersonic MiB," in *Proceedings of the IEEE International Workshop on Genomic Signal Processing and Statistics (GENSIPS '13)*, pp. 86–87, Houston, Tex, USA, November 2013.

[72] B. Wajid, E. Serpedin, M. Nounou, and H. Nounou, "MiB: a comparative assembly processing pipeline," in *Proceedings of the IEEE International Workshop on Genomic Signal Processing and Statistics (GENSIPS '12)*, pp. 86–89, IEEE, Washington, DC, USA, December 2012.

[73] B. Wajid, E. Serpedin, M. Nounou, and H. Nounou, "A modular approach to reference assisted sequence assembly," *International Journal of Computational Biology and Drug Design*, vol. 8, no. 3, 2015.

[74] M. Boetzer, C. V. Henkel, H. J. Jansen, D. Butler, and W. Pirovano, "Scaffolding pre-assembled contigs using SSPACE," *Bioinformatics*, vol. 27, no. 4, pp. 578–579, 2011.

[75] L. Salmela, V. Mäkinen, N. Välimäki, J. Ylinen, and E. Ukkonen, "Fast scaffolding with small independent mixed integer programs," *Bioinformatics*, vol. 27, no. 23, Article ID btr562, pp. 3259–3265, 2011.

[76] K. Worley, "Improving draft genome assemblies using next-gen data with gap-filling and scaffolding assembly tools," in *Proceedings of the Plant and Animal Genome XX Conference*, Plant and Animal Genome, January 2012.

[77] K. Sahlin, F. Vezzi, B. Nystedt, J. Lundeberg, and L. Arvestad, "BESST-efficient scaffolding of large fragmented assemblies," *BMC Bioinformatics*, vol. 15, article 281, 2014.

[78] N. Donmez and M. Brudno, "SCARPA: Scaffolding reads with practical algorithms," *Bioinformatics*, vol. 29, no. 4, pp. 428–434, 2013.

[79] M. Ghodsi, C. M. Hill, I. Astrovskaya et al., "*De novo* likelihood-based measures for comparing genome assemblies," *BMC Research Notes*, vol. 6, no. 1, article 334, 2013.

[80] A. Rahman and L. Pachter, "CGAL: computing genome assembly likelihoods," *Genome Biology*, vol. 14, no. 1, article R8, 2013.

[81] S. C. Clark, R. Egan, P. I. Frazier, and Z. Wang, "ALE: a generic assembly likelihood evaluation framework for assessing the accuracy of genome and metagenome assemblies," *Bioinformatics*, vol. 29, no. 4, pp. 435–443, 2013.

[82] W. Huang and G. Marth, "EagleView: a genome assembly viewer for next-generation sequencing technologies," *Genome Research*, vol. 18, no. 9, pp. 1538–1543, 2008.

[83] M. Laczik, E. Tukacs, B. Uzonyi et al., "Geno viewer, a SAM/BAM viewer tool," *Bioinformation*, vol. 8, no. 2, pp. 107–109, 2012.

[84] S. Andrews, *SeqMonk*, 2007.

[85] K. Popendorf and Y. Sakakibara, "SAMSCOPE: an OpenGL-based real-time interactive scale-free SAM viewer," *Bioinformatics*, vol. 28, no. 9, pp. 1276–1277, 2012.

[86] I. Milne, M. Bayer, L. Cardle et al., "Tablet-next generation sequence assembly visualization," *Bioinformatics*, vol. 26, no. 3, pp. 401–402, 2009.

[87] H. M. Manske and D. P. Kwiatkowski, "LookSeq: a browser-based viewer for deep sequencing data," *Genome Research*, vol. 19, no. 11, pp. 2125–2132, 2009.

[88] M. C. Schatz, A. M. Phillippy, D. D. Sommer et al., "Hawkeye and AMOS: visualizing and assessing the quality of genome assemblies," *Briefings in Bioinformatics*, vol. 14, no. 2, pp. 213–224, 2013.