Comparison of de novo Genome Assembly Software Tools, and Second and Third Generation Sequencing Technology

Wenyuan Yu, E-mail: wyu@uoregon.edu

Table of Contents

1. Abstract

In this term paper, I will basically discuss two topics of genome assembly. First, two new assemblers, SGA (String Graph Assembler) and Fermi, which are developed in recent two years, are introduced. And then I will give general introduction of recent popular de novo assemblers,

such as AbySS, Velvet, SOAP de novo and ALLPATHS-LG *etc*. The performance of eight commonly used assemblers was evaluated by computational time and memory cost, assembly accuracy, completeness and size distribution of assembled contigs. This information will assist researchers in selecting a well-suited assembler and offer essential information for the improvement of existing assemblers or the developing of novel assemblers.

The second topic is about Next-Generation Sequencing (NGS) Systems (Second Generation Sequencing Systems) and the new development of third-generation sequencing technologies. The three main NGS systems 454 GS Junior (Roche), MiSeq (Illumina) and Ion Torrent PGM (Life Technologies) will be briefly reviewed. The new technology and advantages of third-generation sequencing will be discussed.

2. De Novo Genome Assembly

2.1 Introduction

The next-generation sequencing (NGS) technologies have been evolving rapidly in recent years; however, the analysis of short reads datasets after sequencing is still a tough task. One of the biggest challenges for the analysis of high throughput sequencing reads is the whole genome assembly. This process of genome assembly is complicated by the different read lengths, read counts, and error profiles that are produced by different NGS technologies. Sequencing with longer reads is a potential solution, while it becomes impractical with limit current of sequencing technology. Another big challenge for the assembly of short reads is the intensive computational time requirement.[1]

The need to assemble genomes from NGS data has led to the development of many novel assembly software. In the next paragraphs, I will introduce several newly developed or popular de novo assemblers.

2.2 New de novo assemblers

2.2.1 GSA Assembler

Several other assemblers have been developed in recent two years, such as SGA (String Graph Assembler) and Fermi. SGA is based on using the FM-index derived from the compressed Burrows-Wheeler transform. Most de novo assemblers rely on de Bruijn graphs; GSA uses the overlap-based string graph model of assembly, and is simply parallelizable.

GSA demonstrates the error correction and assembly performance of SGA on 1.2 billion sequence reads from a human genome, which can be assembled using 54 GB of memory. The resulting contigs are highly accurate and contiguous, while covering 95% of the reference genome (excluding contigs < 200 bp in length). GSA gives similar results as SOAP de novo in the overall assembly accuracy for *C. elegans N2* and whole human genome assembly (Figure 1). Because of the low memory requirements and parallelization without requiring inter-process communication, SGA provides the first practical assembler for a mammalian-sized genome on a low-end computing cluster.[2]

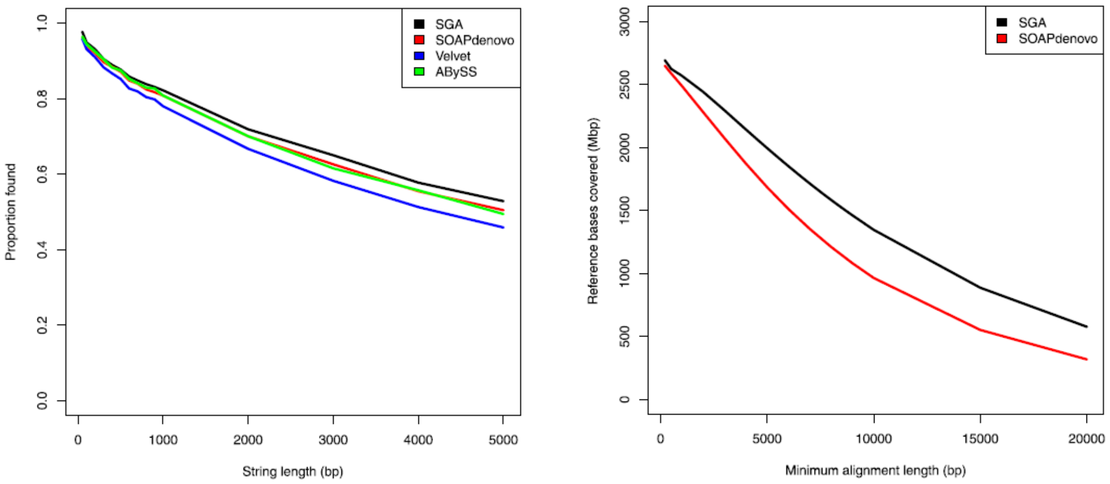


Figure 1. Left: Reference string coverage analysis for the *C. elegans N2* assembly, the proportion of the strings found in the SGA, Velvet, ABySS, and SOAPdenovo; Right: the amount of the human reference genome covered by a contig as a function of the minimum contig alignment length. (Figures are from reference 1)

2.2.2 Fermi Assembler

Fermi is another string graph assembler and has a similar workflow with SGA. Fermi is a de novo assembler with a particular focus on assembling Illumina short sequence reads from a mammal-sized genome. In addition to the role of a typical assembler, fermi also aims to preserve heterozygotes which are often collapsed by other assemblers. Its ultimate goal is to find a minimal set of unitigs to represent all the information in raw reads. Fermi follows the overlap-layout-consensus paradigm and uses the FM-DNA-index (FMD-index) as the key data structure. As a typical de novo assembler, fermi tends to produce contigs with slightly longer N50. However, the major weakness of fermi is the high misassembly rate. Although fermi provides a tool to fix misassemblies by using paired-end reads to achieve an accuracy comparable to other assemblers, this is not a favorable solution.[3]

2.3 General Introduction of Popular de novo Assemblers

Over twenty academic de novo genome assemblers, each possessing its own range of application, are developed for short reads datasets from different sequencing (Figure 2). The comparison and evaluation of tools for de novo assembly of genome sequence will be discussed.[4]

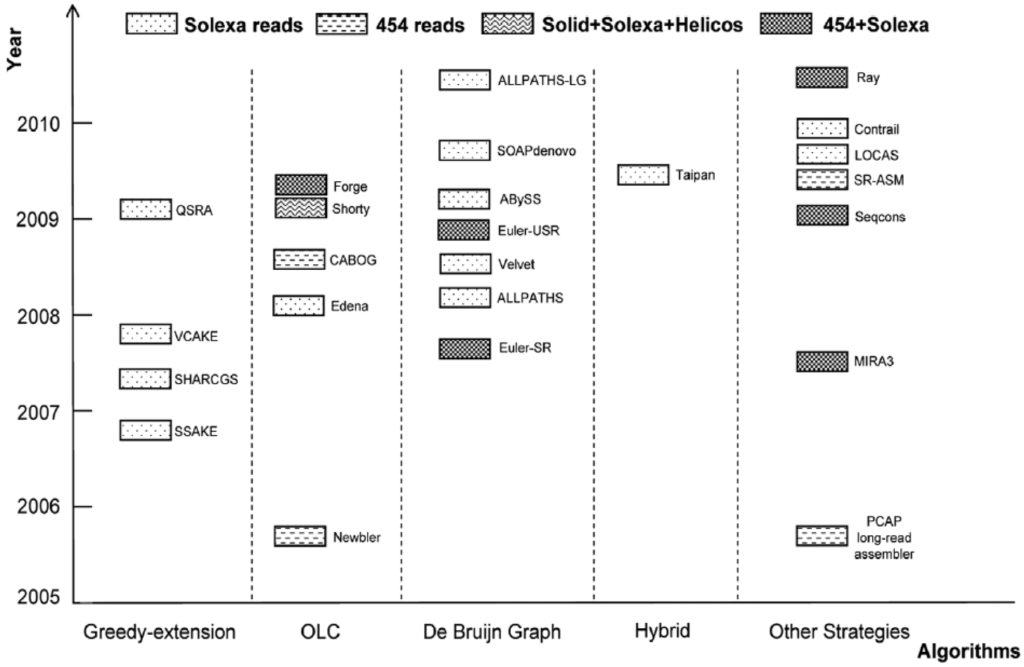


Figure 2. Overview of de novo short reads assemblers. (Figure is from Reference 4)

The strategies applied in short reads assembly can be divided in the following categories:

(1) Greedy extension: SSAKE, VCAKE, QSRA and SHARCGS;

(2) Overlaplayout-consensus (OLC): Edena, Cabog, Forge and Shorty;

(3) De Bruijn: ALLPATHS, SOAP de novo, ABySS, Velvet and APLLPATHS *etc*.;

(4) Hybrid: TAIPAN;

(5) Other Strategies: Seqcons, LOCAS, RAY, Contrail, SR-ASM.

Currently, existing data structure for assemblers can classified into two categories: string-based model and graph-based model. Greedy-extension is the implementation of string-based method. OLC and *De Bruijn* graph are two different graph-based approaches. Taipan was proposed as the hybrid of string-based and graph based approaches, with the dominative feature – the exceedingly short runtime.[5]

String-based assemblers, implemented with Greedy-extension algorithm, are mainly reported for the assembly of small genomes, while the latter ones are designed aiming at handling complex genomes. In terms of the computational time, maximum random access memory (RAM) occupancy, assembly accuracy and integrity, study has indicated that string-based assemblers, overlap-layout-consensus (OLC) assemblers are well-suited for very short reads and longer reads of small genomes respectively. For large datasets of more than hundred millions of short reads, De Bruijn graph-based assemblers would be more appropriate.

2.4 Comparison of several de novo assemblers

In order to compare different de novo assemblers, eight short reads assemblers (SSAKE, VCAKE, QSRA, SHARCGS, Edena, Velvet, SOAP de novo and Taipan), representing four various assembly strategies, were benchmarked against two types of simulated short reads datasets (SE, PE) derived from four different genomes (*C. elegans*, *Yeast, E.coli, and Swinepox*). In the absence of a high-quality reference genome, new genome assemblies are often evaluated on the basis of the number of scaffolds and contigs required to represent the genome, the proportion of reads that can be assembled, the absolute length of contigs and scaffolds, and the length of contigs and scaffolds relative to the size of the genome. The assemblers' performance information is evaluated by computational time and memory cost, assembly accuracy, completeness and size distribution of assembled contigs. Each assembler is applied to handle datasets with different data sizes.

2.4.1 Computational running time and maximum memory cost

The computational time of the assembly process is determined by both the dataset complexity and the assembly strategy. The information about running times, maximum memory occupancies for the eight assemblers applied to different datasets is illustrated in Figures 3.

From figure 3, for string-based assembler (SSAKE, VCAKE, QSRA and SHARCGS), the time and memory cost is approximately proportionate to dataset size. Edena has two running modes: strict and nonstrict modes. For the strict mode, fewer but more accurate contigs are generated, while nonstrict mode acts on the contrary. Compared with string-based tools, Edena is superior in terms of time and RAM utilization.

Velvet and SOAPdenovo are another graph-based method. Both of them run assembly tasks with small computational time and memory usage. SOAPdenovo runs in an extreme speed as the exploitation of threads parallelization, but may perform not well enough for small datasets due to the initial task allocation.

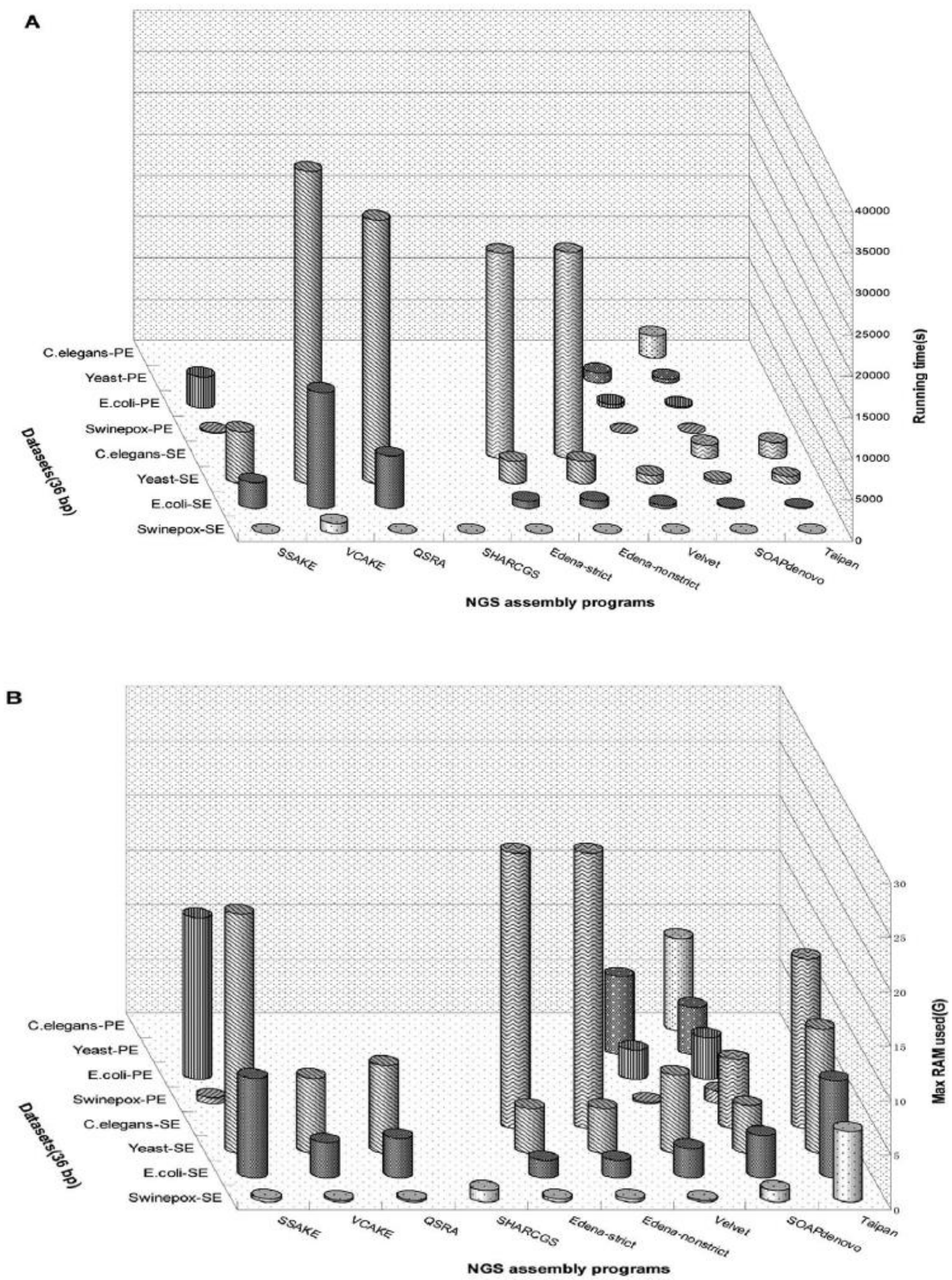


Figure 3. Computational running time and maximum memory occupancy of 36-mer short reads assembly procedures. (A) the computational times of each assembler for different datasets. (B) the maximum RAM used during the assembly process. (Figure is from Reference 4)

Taipan was proposed as the hybrid of string-based and graph-based approaches, with the dominative feature – the exceedingly short runtime. Nevertheless, the minimum RAM of computer to execute the assembler is high and the requirement for memory grows slowly with the increase of dataset size.

2.4.2 Assembly accuracy

The assembly accuracy and integrity is another consideration for the evaluation of the short reads assemblers. Obviously, contigs with high fidelity and genome coverage are our expectation. Different assemblers have their own performance.

String-based assemblers, such as VCAKE and SHARCGS, performed in rivalry with the latest version of SSAKE while QSRA could only generate less precise and lower coverage contigs. What deserves to be mentioned is that Edena, as an assembler based on the overlap-layout-consensus algorithm, had a quite surpassing performance on various datasets. However, contigs produced from two De Bruijn graph-based assemblers, especially SOAPdenovo, were of lower accuracy, but with comparable genome coverage to string-based software. Nevertheless, when handling dataset of huge size, such as short reads from *C.elegans* genome, SOAPdenovo had similar performance as Edena. This result can be elucidated as following: for De Bruijn graph-based method, certain proportion of base errors are incorporated into contigs during the construction of graph with k mers generated from input short reads, this process then generate less precise contigs.[4]

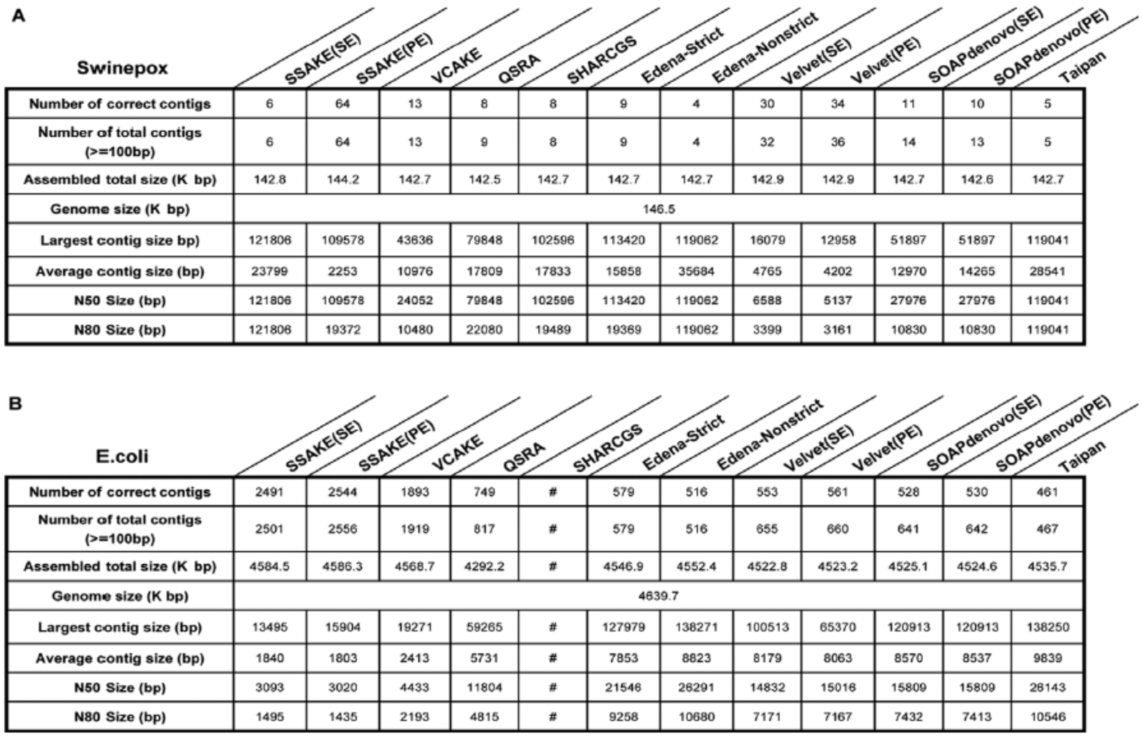
Taipan was capable to generate sequences of high accuracy and genome coverage as string-based assembler for small datasets, but performed poorly for the assembly of large genome dataset. PE reads is superior to SE reads in terms of resolution for repetitive elements, which is in consistent with previous study.

2.4.3 Completeness and size distribution of assembled contigs

Under ideal condition, only one contig that matches the whole genome sequence perfectly could be generated from each assembly procedure. Practically, the contigs generated by different assembly procedures are separated by gaps for the presence of repetitive fragments.

The most commonly used metric is N50. N50 is calculated by summing all sequence lengths, starting with the longest, and observing the length that takes the sum length past 50% of the total assembly length. But this metric may not accurately reflect the quality of an assembly.[6]

Thus, both N50 and N80 size represent the maximum read length for which all contigs greater than or equal to the threshold covered 50% or 80% of the reference genome. Some statistics of for assembled contigs of 36-mer short reads of *E.coli* and *Swinepox* are shown in Figure 4.



A

Swinepox	SSAKE(SE)	SSAKE(PE)	VCAKE	QSRA	SHARCGS	Edena-Strict	Edena-Nonstrict	Velvet(SE)	Velvet(PE)	SOAPdenovo(SE)	SOAPdenovo(PE)	Taipan
Number of correct contigs	6	64	13	8	8	9	4	30	34	11	10	5
Number of total contigs (>=100bp)	6	64	13	9	8	9	4	32	36	14	13	5
Assembled total size (K bp)	142.8	144.2	142.7	142.5	142.7	142.7	142.7	142.9	142.9	142.7	142.6	142.7
Genome size (K bp)	146.5											
Largest contig size bp)	121806	109578	43636	79848	102596	113420	119062	16079	12958	51897	51897	119041
Average contig size (bp)	23799	2253	10976	17809	17833	15858	35684	4765	4202	12970	14265	28541
N50 Size (bp)	121806	109578	24052	79848	102596	113420	119062	6588	5137	27976	27976	119041
N80 Size (bp)	121806	19372	10480	22080	19489	19369	119062	3399	3161	10830	10830	119041

B

E.coli	SSAKE(SE)	SSAKE(PE)	VCAKE	QSRA	SHARCGS	Edena-Strict	Edena-Nonstrict	Velvet(SE)	Velvet(PE)	SOAPdenovo(SE)	SOAPdenovo(PE)	Taipan
Number of correct contigs	2491	2544	1893	749	#	579	516	553	561	528	530	461
Number of total contigs (>=100bp)	2501	2556	1919	817	#	579	516	655	660	641	642	467
Assembled total size (K bp)	4584.5	4586.3	4568.7	4292.2	#	4546.9	4552.4	4522.8	4523.2	4525.1	4524.6	4535.7
Genome size (K bp)	4639.7											
Largest contig size (bp)	13495	15904	19271	59265	#	127979	138271	100513	65370	120913	120913	138250
Average contig size (bp)	1840	1803	2413	5731	#	7853	8823	8179	8063	8570	8537	9839
N50 Size (bp)	3093	3020	4433	11804	#	21546	26291	14832	15016	15809	15809	26143
N80 Size (bp)	1495	1435	2193	4815	#	9258	10680	7171	7167	7432	7413	10546

Figure 4. Statistics for assembled contigs of 36-mer short reads (Figure is from Reference 4).

The result shows that for dataset of very small size, string-based assemblers produced fewer but longer reads than De Bruijn graph based tools. However, it became reverse when the size of dataset increases. Edena, the OLC assembler, could assemble short reads into relatively long contigs for various datasets. Taipan, as a hybrid assembly tool, had better performance than Edena for small datasets. For De Bruijn assemblers, Velvet produced better assembly result than SOAPdenovo when assembly of 75-mer short reads datasets, because of the wider range of K value to be chosen in Velvet. New version of SOAPdenovo (SOAPdenovo2) which comes out in 2013, greatly surpasses its predecessor SOAPdenovo and is competitive to other assemblers on both assembly length and accuracy.[7]

2.5 Conclusion

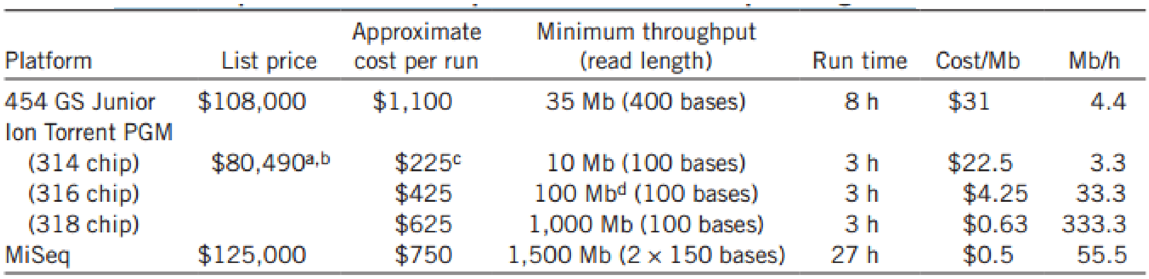
Considering the computational time, maximum random access memory (RAM) occupancy, assembly accuracy and integrity, the study indicate that string-based assemblers, overlap-layout-consensus (OLC) assemblers are well-suited for very short reads and longer reads of small genomes respectively. For large datasets of more than hundred millions of short reads, De Bruijn graph-based assemblers would be more appropriate. In terms of software implementation, string-based assemblers are superior to graph-based ones, of which SOAPdenovo is complex for the creation of configuration file.[4]

3. Second Generation Sequencing and Third Generation Sequencer.

In the second part of this paper, I will briefly talk about Next-Generation Sequencing (Second Generation Sequencing) Systems and the newly developed third Generation Sequencer.

3.1 Second Generation Sequencer

NGS systems are typically represented by the 454 GS Junior (Roche), MiSeq (Illumina) and Ion Torrent PGM (Life Technologies).[8] Figure 5 shows that price comparison of three main three NGS system.



Platform	List price	Approximate cost per run	Minimum throughput (read length)	Run time	Cost/Mb	Mb/h
454 GS Junior	$108,000	$1,100	35 Mb (400 bases)	8 h	$31	4.4
Ion Torrent PGM						
(314 chip)	$80,490[a,b]	$225[c]	10 Mb (100 bases)	3 h	$22.5	3.3
(316 chip)		$425	100 Mb[d] (100 bases)	3 h	$4.25	33.3
(318 chip)		$625	1,000 Mb (100 bases)	3 h	$0.63	333.3
MiSeq	$125,000	$750	1,500 Mb (2 × 150 bases)	27 h	$0.5	55.5

Figure 5. Price comparison of benchtop instruments and sequencing runs (Figure is from reference 9).

In order to compare the performance of these three instruments, the researchers in the research article compared the three sequencers by sequencing an isolate of Escherichia coli O104:H4.[9] The result showed that the MiSeq had the highest throughput per run (1.6 Gb/ run, 60 Mb/h) and lowest error rates. The 454 GS Junior generated the longest reads (up to 600 bases) and most contiguous assemblies but had the lowest throughput (70 Mb/run, 9 Mb/h). Run in 100-bp mode, the Ion Torrent PGM had the highest throughput (80–100 Mb/h). Unlike the MiSeq, the Ion Torrent PGM and 454 GS Junior both produced homopolymer-associated indel errors.[9]

3.2 Third Generation Sequencer

While the increasing usage and new modification in next generation sequencing, the third generation sequencing is coming out with new insight in the sequencing. Third-generation sequencing has two main characteristics. First, PCR is not needed before sequencing, which shortens DNA preparation time for sequencing. Second, the signal is captured in real time, which means that the signal, no matter whether it is fluorescent (Pacbio) or electric current (Nanopore), is monitored during the enzymatic reaction of adding nucleotide in the complementary strand.[8]

Comparing to second generation, PacBio RS (the first sequencer launched by PacBio) has several advantages. First, the sample preparation is very fast; it takes 4 to 6 hours instead of days.

Also it does not need PCR step in the preparation step, which reduces bias and error caused by PCR. Second, the turnover rate is quite fast; runs are finished within a day. Third, the average read length is 1300 bp, which is longer than that of any second-generation sequencing technology. Although the throughput of the PacBioRS is lower than second-generation sequencer, this technology is quite useful for clinical laboratories, especially for microbiology research.

Nanopore sequencing is another method of the third generation sequencing. Nanopore sequencing possesses a number of fruitful advantages over existing commercialized next-generation sequencing technologies. Firstly, it potentially reaches long read length > 5 kbp with speed 1 bp/ns. Moreover, detection of bases is fluorescent tag-free. Thirdly, except the use of exonuclease for holding up ssDNA and nucleotide cleavage, involvement of enzyme is remarkably obviated in nanopore sequencing. This implies that nanopore sequencing is less sensitive to temperature throughout the sequencing reaction and reliable outcome can bemaintained. Fourthly, instead of sequencing DNA during polymerization, single DNA strands are sequenced through nanopore by means of DNA strand depolymerization.[8]

4. Reference

1. Miller JR, Koren S, Sutton G (2010) Assembly algorithms for next-generation sequencing data. *Genomics* 95: 315–327.

2. Simpson JT, Durbin R: Efficient de novo assembly of large genomes using compressed data structures. *Genome Res* 2011, 22:549–556

3. Li H. Exploring single-sample SNP and INDEL calling with whole-genome de novo assembly. *Bioinformatics* 2012, 28:1838–1844

4. Zhang W, Chen J, Yang Y, Tang Y, Shang J, et al. A Practical Comparison of De Novo Genome Assembly Software Tools for Next-Generation Sequencing Technologies. *PLoS ONE* 2011, 6: e17915

5. Schmidt B, Sinha R, Beresford-Smith B, Puglisi SJ A fast hybrid short read fragment assembly algorithm. *Bioinformatics* 2009, 25: 2279–2280.

6. Parra, G., Bradnam, K., Ning, Z., Keane, T. & Korf, I. Nucleic Acids Res. 2009, 37, 289–297.

7. Luo et al.: SOAPdenovo2: an empirically improved memory-efficient short-read de novo assembler. *GigaScience* 2012 1:18.

8. Lin Liu, Yinhu Li, Siliang Li, Ni Hu, Yimin He, Ray Pong, Danni Lin, Lihua Lu, andMaggie Law Comparison of Next-Generation Sequencing Systems. *Journal of Biomedicine and Biotechnology* 2012, Article ID 251364

9. Nicholas J Loman, Raju V Misra, Timothy J Dallman, Chrystala Constantinidou, Saheer E Gharbia, John Wain Mark J PallenPerformance comparison of benchtop high-throughput sequencing platforms *Nature Biotechnology* 2012, 30,434–439