Vol.2011-BIO-24 No.4
2011/3/10

An Effective Approach for Assembling Very Short Reads from Next Generation Sequencer

Wisnu Ananta Kusuma[†], Takashi Ishida[†], and Yutaka Akiyama[†]

de novo DNA sequence assembly is very important in genome sequence analysis. In this paper, we deeply investigated two of the major approaches for *de novo* DNA sequence assembly of very short reads: overlap-layout-consensus (OLC) and Eulerian path using a de Bruijn graph. We proposed a new assembling technique by combining OLC and Eulerian path in hierarchical process. The contigs yielded by these two approaches are treated as reads and are assembled again to yield longer contigs than the previous contigs. We tested the combined approach by using three Illumina very-short-read datasets and four error-free reads simulated datasets. As results, the combined approach yielded longer contigs, in term of N50 and maximum contigs, than OLC or Eulerian path approach alone.

1. Introduction

The sequencing of DNA is very important in genomics. Currently, most technologies for sequencing genomes depend on the shotgun method. In this technique, genomes are randomly cut into many fragments, and a computer programs is required to reconstruct the DNA sequence. Fragments of DNA sequences can be assembled in two ways: by mapping the DNA sequences to a reference sequence or by using *de novo* DNA sequence assembling techniques, if there are no reference sequences because of newly sequenced organisms.

De novo DNA sequence assembly generally falls into two main approaches, namely, overlap-layout-consensus (OLC) and Eulerian path using a de Bruijn graph. The OLC approach is very intuitive approach that represents the sequence assembly problem as an overlap graph. In this graph, each node represents a read, and each edge represents an overlap between two reads. Celera[1)], ARACHNE[2)], PCAP[3)], Phusion[4),] and Edena[5)] was developed based on this approach. Actually, OLC has been proved to be suitable for assembling the long reads (500 bp) produced by Sanger sequencing[1)]. However, it is not suitable for assembling very short reads (35-50 bp)[7)] produced by a next generation sequencer. Applying the OLC approach to very short reads makes the overlapping stage more difficult. Repeats are much more prevalent in very short reads because the overlap region is shorter. The higher frequency of repeats will increase the probability of finding the same substring, not just in the overlap region but also in other parts of the read. Thus, repeats cause ambiguity and tend to generate misassembled contigs.

The difficulty of handling repeats that commonly faced by assemblers based on OLC has motivated other researchers to find an alternative approach. Pevzner[6)] attempted to solve this problem by introducing a Eulerian path approach using a de Bruijn graph. In this approach, elements are not organized around reads but around words of k-nucleotide repeats, called k-mers. This approach transposes the difficulties of the OLC problem in dealing with sequence assembly for very short reads into a Eulerian path problem by using a de Bruijn graph. Some of the well-known assemblers based on the Eulerian path approach include EULER[6)], Velvet[7)], and ALLPATHS[8)].

An assembler is said to have a better performance if it yields fewer contigs with a longer maximum contig length and N50 value than the other assemblers. Thus, one of the important objectives of developing new technique for *de novo* DNA assembly is to obtain higher value in the term of N50 value and the maximum contigs length. The N50 size is the size of the smallest contig such that the total length of all contigs greater than this size at least one half of the total genome size. Velvet, one of most popular assembler based on Eulerian path showed the effectiveness of Eulerian path approach in assembling very short reads. Velvet could yield longer contigs in term of N50 value and maximum contigs than the other assemblers. However, later, Edena assembler (based on OLC) showed a performance similar to those of the Velvet assembler in assembling Illumina very short reads. Actually, Velvet and Edena use different approach include: different data representation and different strategies to simplify graph, to remove sequencing errors and to find the shortest paths as solutions. Intuitively, applying different approach will obtain different contigs which may cover different region in the genome. The author of Edena also indicated this possibility that we can yield longer contigs by using two assemblers[5)]. However, no researcher have already checked and elaborated it.

In this research, we proposed a new assembling technique by combining OLC and Eulerian path in hierarchical process. Contigs yielded by two assemblers with different approaches will be treated as reads and will be assembled again to yield longer contigs than previous contigs. This research was carried out to assess the feasibility of yielding longer contigs in *de novo* sequence assembly for very short reads from next generation sequencer.

2. *de novo* DNA Sequence Assembly Approach

The problem of assembling DNA sequences can be formulated into the problem of finding the shortest common superstring (SCS), which is known to be NP-hard[9)]. Suppose we have a DNA sequence from an unknown source of A= $a_1, a_2,..,a_L$. Shotgun sequencing of A produces a set of reads (or fragments) F = $\{f_1, f_2,\ldots f_R\}$ that are sequences over the alphabet

[†] Graduate School of Information Science and Engineering, Tokyo Institute of Technology

$\sum$ = {A, C, G, T}. To reconstruct the sequence A from the set of its fragments F = {f_1, f_2,…f_R}, we need to find the minimum string length that is a superstring of every $f_i \in$ F.

The definition of a superstring can be illustrated as follows[10]. Let $s_1 = a_1 \ldots a_r$ and $s_2 = b_1 \ldots b_s$ be strings over some finite alphabet $\sum$. We say that s_2 is a superstring of s_1 if there is an integer $i \in [0, s - r]$ such that $a_j = b_{i+j}$ for $i \leq j \leq r$.

SCS is a simple model for formulating the real problem of genome assembly. The SCS is formed based on the assumption that every read must be present in the original genome. Thus, the original genome should be the shortest sequence that contains every read as a substring[11]. There are two major approaches to solve the SCS problem: overlap-layout-consensus (OLC) and Eulerian path using a de Bruijn graph.

2.1 Overlap Layout Consensus (OLC) approach

The OLC approach consists of three steps: overlap, layout, and consensus. In the overlap step, we first find potentially overlapping reads using a greedy approach. This information is then used to construct an overlap graph by the following procedure: construct a graph with n vertices, representing the n strings (reads) s_1, s_2,…, s_n, and insert edges of length overlap (s_i, s_j) between the vertices s_i and s_j. For this purpose, Medvedev *et al* [11] define overlap in the general form as follows: let *v* and *w* be two strings over the alphabet $\sum$ = {A, C, G, T}. If there exists a maximal length non-empty string *z* that is a prefix of *w* and a suffix of *v*, then *w* overlaps *v*. This definition is not symmetric. |*z*| is the length of the overlap. If |*z*| = 0, *w* does not overlap *v*.

During the layout stage, the overlap graph is analyzed to find the path in the overlap graph that visits every vertex exactly once. It is a Hamiltonian path problem, which is known to be NP-hard. However, this overlap graph formulation is in fact not suitable for finding the single path that represents the shortest superstring[12]. Therefore; current genome assemblers still produce many contigs as their outputs. Furthermore, in the layout stage, the set of contigs is merged to yield supercontigs. For this purpose, we need information on the mate pair lengths to estimate the distance between two contigs that are to be merged. The final step in the OLC strategy is consensus. The goal of this step is to determine the DNA sequence by aligning all the reads covering the genome. The consensus sequence is determined by vote using quality values.

2.2 Eulerian path approach

Currently, the Eulerian path approach introduced by Pevzner[6] is very popular. It adopts de Bruijn graphs to assemble the sequence by organizing (k-1)-mers as vertices and k-mers as edges. Thus, any walk that contains all of the reads as subwalks represents a valid assembly. Let S = {s_1, …, s_n} be a set of strings over an alphabet $\sum$ = {A, C, G, T} and let G = B_k(S) be the de Bruijn graph of S for some k. The string si corresponds to walks in B_k(S) via the function w(s) = s[1…k] →s[2..k+1] → . . . → s[| s | - k + 1, | s |]. A walk is called a superwalk of G if, for all i, it contains w(s_i) as subwalk. Thus, a superwalk represents a valid assembly of the reads into a genome. Formally, given a set string S, as defined above, and positive integer k, the de Bruijn Superwalk Problem (BSP) is to find the minimum length superwalk in B_k(S). This approach can simplify the complexity of the layout problem in the OLC approach into an Eulerian path problem that can be solved efficiently.

2.3 The real problem in DNA sequence assembly

Actually there are three important problems in DNA assembly: unknown orientations, the presence of repeats, and the existence of sequencing errors. In this study, we deeply investigated how these problems were handled by each approach. We used two assemblers: Edena and Velvet as a case. We choose Edena and Velvet because both of them have showed the best performance in assembling very short reads.

Edena adopts a bidirected overlap graph to deal with unknown orientation problem. In this graph, each node correspond to double stranded reads (read and its reverse complement), and each overlap (edge) has an orientation at both endpoints[11]. There are three possible ways that two double-stranded reads can overlap (Figure 1). The weight of a graph can be assigned as the sum of the weights of its edges. Additionally, if a weighted bidirected graph is known, by employing the Chinese Postman Problem (CPP), we can find a cyclical walk that traverses every edge at least once. On the other hand, Velvet adopts a bi-directed de Bruijn graph, to overcome the unknown orientation problem. In a bidirected de Bruijn graph (Figure 2), the nodes of the graph will be all of the possible (k-1)-molecules. A k-molecule is defined as a k-mer and its complement. The shortest assembly of the DNA molecule with the given k-molecule-spectrum can be found by applying the Chinese Postman tour of the resulting bidirected de Bruijn graph.

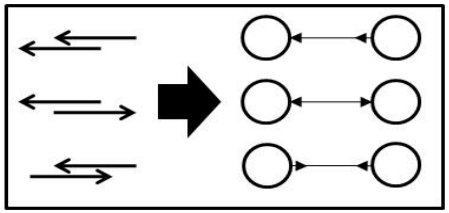

Figure 1 this figure describe three possible ways that two double-stranded reads can overlap. It corresponds to the three types of edges. Suppose we have two reads r_1 and r_2. Each read can be in either of two orientations; oriented to the left or to the right. The three possible overlaps are: i) both strands point in the same direction (both reads can point left, or both can point right), ii) r_1 points left and r_2 points right, iii) r_1 points right and r_2 points left. There are also three corresponding types of bidirected edges. The left node corresponds to the lower read. Note that the arrow points into a node if and only if the overlap covers the start (5') of the read[11].

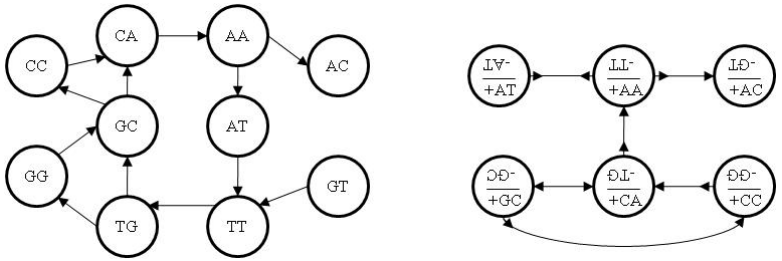

Figure 2 From the k-molecule spectrum {ATT/AAT, TTG/CAA, TGC/GCA, GCC/GGC, CCA/TGG, CAA/TTG, AAC/GTT}, we can generate a de Bruijn Graph (on the left) and a bidiriected de Bruijn Graph (on the right)[11)]

Furthermore, when dealing with repeats, that is, multiple copies of identical substrings at different positions in the DNA, the OLC approach faces a significant problem. Very short reads make repeats much more prevalent in the OLC. This is because the overlap region is shorter. The presence of repeats will increase the possibility of finding the same substring, not just in the overlap region but also in other parts of the read. Thus, repeats cause ambiguity and tend to generate misassembled contigs. However, Edena has successfully demonstrated that the OLC approach can also yield significant contigs by employing a suffix array to perform the overlapping phase[5)]. For the Eulerian path approach, repeats can be handled intrinsically using a de Bruijn graph.

To deal with the third problem, the presence of sequencing errors, both Edena and Velvet use a topological approach. This approach is implemented after constructing a graph. In Edena, sequencing errors are eliminated by removing short dead-end paths and small bubbles in the graph. In Velvet, sequencing errors are eliminated by removing erroneous edges and bubbles.

Moreover, the functional features of DNA sequence assembly also depend on the functionality of the approach used. Assemblers based on OLC usually employ three processes: constructing overlap graphs, finding Hamiltonian paths to yield contigs, and performing the consensus step. On the other hand, the Eulerian path approach usually involves two main functions, such as constructing a de Bruijn graphs and generating contigs by finding Eulerian paths. Because different data representations are used and because different strategies for finding the shortest path are used, the contigs produced by two assemblers that employ different approaches may cover different regions in a genome. If the contigs produced by the two assemblers have overlapping regions between them, these two assemblers may be complementary to each other and yield longer contigs[5)].

2.4 A combined approach

In this research, we proposed to assemble very short reads hierarchically by combining OLC and Eulerian path. The idea of this approach is simple. The contigs yielded by these two approaches can be treated again as reads and can be assembled to yield longer contigs than the previous contigs. In this case, we should use two different approaches because, as described in Table 1, both assemblers with different approach use different data representations. Moreover, they also use different strategy to simplify the graph and to find the shortest for producing contigs. Thus, the contigs yielded by these two different approaches may cover different regions in the genome and have overlapping regions between them. As a result, it is possible to assemble these two sets of contigs to yield longer contigs.

Table 1 Comparison of data representation and strategies used by Edena and Velvet

	Edena	Velvet
Basic data representation	Overlap graph	De Bruijn graph
Handling unknown orientation	Bi-directed overlap graph	Bi-directed de Bruijn graph
Handling repeats	Handled in overlap phase by using suffix array	Intrinsically handled by de Bruijn graph
Handling sequencing errors	using topological approach	using topological approach

We formulate the combined approach in a general form as follows: let $E = \{e_1,\ldots, e_n\}$ be a set of contigs produced by assemblers based on the Eulerian path approach and $O = \{o_1,\ldots, o_n\}$ be a set of contigs produced by assemblers based on the OLC approach. This problem can be solved by finding the region of overlap among the contigs. In the general viewpoint, we can view the object of the problem as finding an ordering of the strings that maximizes the amount of overlap between consecutive strings[10)], as follows:

Let $e_1 = a_1 \ldots a_r$ and $o_1 = b_1 \ldots b_s$ be strings. We define:

$\Phi(e_1, o_1) = \max \{k \geq 0 \mid a_r\text{-}k+i = b_i,\ 1 \leq i \leq k\}$.

We now introduce the combined approach which successfully yields longer contigs. Generally, this approach combines OLC and Eulerian. It is divided by three phase. We applied three assemblers as part of a combined approach, including Edena (based on OLC), Velvet (based on the Eulerian path), and Minimus (based on OLC)[13)], as implemented by Hernandez[5)].

In the first phase, we assembled very short reads with Velvet. Velvet is started by hashing reads according to predefined length to construct de Bruijn graph. These graphs are simplified by merging chain of nodes into single nodes. Furthermore, to remove sequencing errors, from the existing graph, some topologies such as tips and bubbles are removed. Contigs produced in this step then are joined by several paired reads to yield final contigs.

In the second phase, the same datasets were assembled by Edena. Firstly, redundant reads will be reduced by indexing reads in a prefix tree. Next, in the overlap step an overlap graphs is constructed. After constructing graph, sequencing errors are eliminated by removing transitive and spurious edges and resolving bubbles. Then, all significant contigs are generated in the layout and consensus step.

Finally, in the third phase, we assembled contigs produced by Velvet and Edena using the Minimus assembler. In this case, the contigs produced by Edena were merged with the contigs produced by Velvet. The merging contigs were then being assembled using Minimus. This simulation combined the Eulerian path and OLC, which are represented by the Velvet and Edena assemblers, respectively. This new approach is readily parallelized[14], which implies that it is potential for assembling large-scale datasets.

3. Experiments

3.1 Dataset and Computer Resource

We now turn to a comparison of the combined approach discussed above with OLC and Eulerian path approach. We examined their performance using three datasets: sequences of *Staphylococcus aureus* strain MW2[5] and *Helicobacter pullorum* NCTC 12824, and *Bacillus anthracis* BA104. The *Helicobacter pullorum* NCTC 12824 and *Bacillus anthracis* BA104 were taken from the NCBI Short Read Archive. The *Staphylococcus aureus* dataset is made up of 3.86 million 35-bp reads. The raw coverage depth is 48X. The *Helicobacter pullorum* dataset is made up of 4.53 million 36-bp reads. The *Bacillus anthracis* BA104 dataset is made up of 7.63 million 50-bp reads.

Moreover, to demonstrate the performance of the combined approach in assembling error-free reads, we used four simulated datasets including the sequences of *Acetobacter pasteurianus* IFO 3283-01, *Rhodobacter erythropolis* PR4, *Streptococcus suis* P17, and *Escherichia coli* 536, which were generated by MetaSim, a sequencing simulator[15]. The sizes of these datasets were 2.91 million, 6.52 million, 2.01 million, and 4.94 million, respectively. These datasets were set as error free. The length of each read was 35 bp with a coverage depth of 10 X.

The programs used in the assembly process were Velvet 0.7.3, Edena 2.0, and Minimus 2.0. All programs were run on a Dual Core AMD Opteron 2.4 Gz CPU supplied with 32 GB of RAM. Notice that, the Velvet assembler is based on the Eulerian path approach. Both Edena and Minimus are based on the OLC approach.

3.2 Results and Discussion

The performance of an assembler is measured by some value such as the number of contigs, the maximum length of the contigs and the N50 size. The N50 size is the size of the smallest contig such that the total length of all contigs greater than this size is at least one half of the total genome size. The N50 can be computed by sorting all of the contigs from largest to smallest and by determining the minimum set of contigs whose sizes total 50% of the entire genome. An assembler is said to have a better performance if it yields fewer contigs with a longer maximum contig length and N50 size than another assembler.

As the first experiment, we considered the real datasets of *Staphylococcus aureus* strain MW2, *Helicobacter pullorum* NCTC 12824 and *Bacillus anthracis* BA104 from the NCBI Short Read Archive. Velvet was used with k-mer values of 21, 23, and 31 for *Staphylococcus aureus* strain MW2, *Helicobacter pullorum* NCTC 12824, and *Bacillus anthracis* BA104, respectively. We used the real datasets to show the assembly quality of each approach in dealing with very short reads that contain sequencing errors. The results showed that using the combined approach with the *Staphylococcus aureus* strain MW2 dataset yielded 890 contigs less than Velvet (1152) and Edena (1175). Moreover, the maximum length and N50 size of the contigs produced by the combined approach were all longer than those of Velvet and Edena. By using the combined approach, the N50 value and the maximum contig length increased to 2 kbp and 10 kbp, respectively (Table 2). The maximum contigs length was 32.74 kbp and the N50 value was 7.4 kbp. This value meant that 50% of all bases were contained in contigs of at least 7.4 kb.

Table 2 Results of the combined approach assembly with an Illumina dataset using *Staphylococcus aureus* strain MW2

Genome	Assembler	Number of contigs	Maximum Length of contigs (kbp)	N50 (kbp)	Total Length of contigs (Mbp)
Staphylococcus aureus	Velvet (k=21)	1152	22.89	5.30	2.78
	Edena	1175	22.89	5.46	2.76
	Combined approach	**890**	**32.73**	**7.40**	2.77
Helicobacter pullorum	Velvet (k=23)	3981	4.38	0.58	1.76
	Edena	4300	4.11	0.35	1.26
	Combined approach	**2570**	**9.20**	**0.86**	1.66
Bacillus anthracis BA104	Velvet (k=31)	3436	58.11	4.70	5.05
	Edena	2203	23.14	5.06	5.03
	Combined approach	1235	**71.13**	**11.80**	5.02

A similar tendency was also indicated through the use of the *Helicobacter pullorum* dataset. The combined approach yielded 2570 contigs less than Velvet (3981) and Edena (4300), and the N50 value and maximum length of contigs increased to 300 bp and 5 kbp, respectively (Table 2). Moreover, the increased maximum length and N50 were very significant when the *Bacillus anthracis* BA104 dataset was used. The N50 value and the maximum lengths of contigs were 6 kbp and 50 kbp longer, respectively, than Velvet and Edena alone.

To deal with the error-free reads represented by the four datasets shown in Table 3, the combined approach could slightly improve the quality of the assembling results. In general, the maximum length of contigs increased by 240 bp, and the N50 value increased by 130 bp (Table 3). This improvement was not significant. This lack of significance may be because the absence of sequencing errors in the reads made both the overlap graph and the de Bruijn graphs simpler. Consequently, most the contigs yielded by Velvet and Edena may cover a similar region. Thus, the combined approach was no longer effective because the overlap region among the contigs produced by Velvet and Edena was not significant.

Table 3 Results of the combined approach assembly with a simulated dataset generated by MetaSim

Genome	Assembler	Number of contigs	Maximum Length of contigs (kbp)	N50 (in kbp)	Total Length of contigs (Mbp)
Acetobacter pasteurianus	Velvet (k=21)	8544	5.66	0.37	2.64
	Edena	9207	5.66	0.30	2.4
	Combined approach	7684	5.66	0.41	2.61
Rhodobacter erythropolis	Velvet (k=19)	6676	3.27	0.48	6.29
	Edena	22,107	2.58	0.29	5.49
	Combined approach	13,218	3.51	0.61	6.23
Streptococcus suis	Velvet (k=21)	6286	5.28	0.35	1.87
	Edena	6672	5.27	0.29	1.68
	Combined approach	5614	5.40	0.39	1.83
Escherichia coli	Velvet (k=19)	11,405	4.37	0.54	4.71
	Edena	16,691	4.81	0.28	4.08
	Combined approach	9871	5.06	0.62	4.66

As we can see from Table 2 and Table 3, the effectiveness of the Velvet in producing significant contigs depends on the value of k. The value of k becomes critical point in constructing de Bruijn graph. Smaller k-mers increase the connectivity of the graph. Meaning that, it will increase sensitivity. On the other hand, smaller k-mers also increase the number of ambiguous repeats in the graph and tend to generate misassembled contigs. Thus, determining the optimal value of k will be important[6]. The result of experiment using Velvet with Staphylococcus aureus strain MW2 supported this statement (Table 4).

Table 4 shows that the highest number of initial node is achieved by k=19 (2,722,855). However, the results of contigs yielded with k=19, for the maximum contig length and the N50 value, were shorter than those with k=21. With k=19, the N50 value and the maximum contig yielded by Velvet were 0.96 kbp and 4.96 kbp. On the other hand, we could yield longer contig with k=21, 5.32 kbp for N50 value and 22.89 kbp for maximum contigs length. It indicated that with k=19 the connectivity of graph was increased. However, the numbers of ambiguous repeats also were increased.

Table 4 the results of experiment using Velvet with lentgh of reads = 35 bp

k	Number of k-mers	Number of nodes of initial graph	Number of contigs	Number of mis-assembled contigs	N50 (kbp)	Max length (kbp)
19	2,929,033	**2,722,855**	4,229	12	**0.96**	**4.96**
21	2,852,510	2,692,679	1,346	0	**5.32**	**22.89**
23	2,824,454	2,666,178	1,152	0	4.42	18.23
25	2,808,408	2,642,353	1,733	0	3.53	19.98
27	2,767,931	2,604,524	3,483	0	1.45	13.77
29	2,585,644	2,544,756	7,271	0	0.50	13.77
31	1,749,151	2,426,956	9,381	2	0.16	20.58

Therefore we also ran the combined approach and Velvet using some values of k with *Helicobacter pullorum* as the real dataset and *Escherichia coli* as the simulated dataset. The main goal of this experiment is to see the performance of the combined approach with varies of k. However, it is important to note that k must be an odd value. By using varies of k, the results of contigs yielded by Velvet, especially for the maximum contig length and the N50 size, are shorter than those of the combined approach (Table 5).

On the other hand, the results in Table 6 showed that the maximum length and N50 size of contigs yielded by the combined approach were almost constant with the various values of k for the Escherichia coli simulated dataset, which represented as an error-free reads. There is a tendency that the combined approach does not depend on the k value when assembling reads that do not contain sequencing errors. Although the results of the combined approach varied

with k when assembling the real dataset such a *Helicobacter pullorum*, as shown in Table 5, the maximum length of contigs and the N50 value were still higher than those of Velvet. These data indicate that we obtained better results using the combined approach with any value for k.

4. Conclusion

In this paper, we have presented an effective approach to yield longer contigs by involving the combination of two main approaches, OLC and the Eulerian path, in order to assemble very short reads from next generation sequencer. The results showed that the combined approach can yield longer contigs than Velvet or Edena alone in the assembly of very short reads produced by an Illumina sequencer and simulated error-free reads. In addition, we obtained significant contigs with any of odd value of k by applying the combined approach.

Table 5 Results of the combined assembly with different values for k using the *Helicobacter pullorum* dataset

	k = 19		**k = 21**		**k = 23**		**k = 25**	
	Velvet	Combined approach	Velvet	Combined approach	Velvet	Combined approach	Velvet	Combined approach
No. of contigs	7047	3178	4968	2369	3981	8311	5174	3238
Max length (kbp)	0.79	4.10	3.10	7.69	4.36	9.20	6.90	9.70
N50 (kbp)	0.07	0.49	0.41	0.95	0.58	0.86	0.39	0.59
Total length (Mbp)	2.24	1.33	1.88	1.68	1.78	1.66	1.66	1.55

Table 6 Results of the combined approach assembly with different values for k using the *Escherichia coli* dataset

	k = 19		**k = 21**		**k = 23**		**k = 25**	
	Velvet	Combined approach	Velvet	Combined approach	Velvet	Combined approach	Velvet	Combined approach
No. of contigs	11,405	9817	15,938	14,066	20,702	9817	20,610	13,798
Max length (kbp)	4.4	5.1	3.8	5.1	4.9	5.3	2.6	5.4
N50 (kbp)	0.54	0.62	0.33	0.38	0.19	0.30	0.12	0.31
Total length (Mbp)	4.7	4.7	4.6	4.5	4.3	4.2	3.5	3.8

References

1) Myers, E. W., Sutton G. G., Delcher, A. L.,et al. 2000, A whole-genome assembly of *Drosophila*, *Science*, 287, 2196-2204.

2) Batzoglou, S., Jaffe, D. B., Stanley, K., et al. 2002, ARACHNE: A whole genome shotgun assembler, Genome Res., 12, 177 – 189.

3) Huang, X., Wang, J., Aluru, S., Yang, S., and Hillier, D. 2003, PCAP: A Whole-Genome Assembly Program, Genome Res., 13, 2164-2170.

4) Mullikin, J. C. and Ning, Z. 2003, The Phusion Assembler, Genome Res., 13, 81-90.

5) Hernandez, D., et. al. 2008, *De novo* bacterial genome sequencing: Millions of very short reads assembled on a desktop computer, Genome Res., 18, 802 – 809.

6) Pevzner, P.A., Tang, H., and Waterman, M.S. 2001, An Euler path approach to DNA fragment assembly, Proc. Natl. Acad. Sci., 98, 9478-9753.

7) Zerbino, D.R. and Birney, E. 2008, Velvet: Algorithms for *de novo* short read assembly using de Bruijn graphs, Genome Res., 18, 821-829.

8) Butler, J., MacCallum, L., Kleber, M., et. al. 2008, ALLPATHS: *De novo* assembly of whole-genome shotgun microreads, Genome Res., 18, 810-820.

Vol.2011-BIO-24 No.4
2011/3/10

9) Gallant, J., Maier, D., and Storer, J. A. 1980, On finding minimal length superstrings, J. Comput. Syst. Sci., 20(1), 50-58

10) Turner, J.S. 1989, Approximation Algorithm for the Shortest Commomn Superstring Problem. Information and Computation, 83, 1-20.

11) Medvedev, P. et. al. 2007, Computability of Models for Sequence Assembly. Algorithms in Bioinformatics. Lecture Notes in Computer Science, Vol. 4645/2007, pp. 289-301

12) Pop, M. 2009, Genome assembly reborn: recent computational challenges. Oxford University Press.

13) Sommer, D. D., et. al. 2007, Minimus: a fast, lightweight genome assembler, BMC Bioinformatics, 8, 64.

14) Kusuma, W. A. and Akiyama, Y. 2010, Design and Simulation of Hybrid de novo DNA Sequence Assembly for Large Eukaryotic Genomes. PDPTA 2010: 104-108

15) Richter DC, Ott F, Auch AF, Schmid R, Huson DH, 2008, MetaSim—A Sequencing Simulator for Genomics and Metagenomics, PLoS ONE, 3, 10.