IPSJ Transactions on Bioinformatics Vol. 4 21–33 (Nov. 2011)

Original Paper

A Combined Approach for *de novo* DNA Sequence Assembly of Very Short Reads

Wisnu Ananta Kusuma,[†1] Takashi Ishida[†1]
and Yutaka Akiyama[†1]

De novo DNA sequence assembly is very important in genome sequence analysis. In this paper, we investigated two of the major approaches for *de novo* DNA sequence assembly of very short reads: overlap-layout-consensus (OLC) and Eulerian path. From that investigation, we developed a new assembly technique by combining the OLC and the Eulerian path methods in a hierarchical process. The contigs yielded by these two approaches were treated as reads and were assembled again to yield longer contigs. We tested our approach using three real very-short-read datasets generated by an Illumina Genome Analyzer and four simulated very-short-read datasets that contained sequencing errors. The sequencing errors were modeled based on Illumina's sequencing technology. As a result, our combined approach yielded longer contigs than those of Edena (OLC) and Velvet (Eulerian path) in various coverage depths and was comparable to SOAPdenovo, in terms of N50 size and maximum contig lengths. The assembly results were also validated by comparing contigs that were produced by assemblers with their reference sequence from an NCBI database. The results show that our approach produces more accurate results than Velvet, Edena, and SOAPdenovo alone. This comparison indicates that our approach is a viable way to assemble very short reads from next generation sequencers.

1. Introduction

The sequencing of DNA is very important in genomics. Determining the whole genome sequence of a species remains an interesting task, particularly after the successful mapping of the whole genomes of both humans[1] and mice[2]. DNA sequencing and assembly also play an important role in many studies, such as those that are evolutionary[3] and metagenomic[4].

Most technologies for sequencing genomes depend on the shotgun method. In this technique, genomes are randomly cut into many fragments, and a computer program (DNA assembler) is required to reconstruct the DNA sequence. Fragments of DNA sequences can be assembled in two ways: by mapping the DNA fragments to a reference sequence[5],[6] 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 methods can be categorized into two main approaches, namely, overlap-layout-consensus (OLC) and Eulerian path using a de Bruijn graph. The OLC approach is a 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[7], ARACHNE[8], PCAP[9], Phusion[10], Edena[11] and Minimus[12] were developed based on this approach. In fact, OLC has been proven to be suitable for assembling the long reads (500 bp) produced by Sanger sequencing[7]. However, it is not suitable for assembling reads produced by next generation sequencers, such as the 454[13], which generates short reads (200 bp), and Illumina[14], which produces very short reads (35–50 bp)[15]. 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 is commonly faced by assemblers based on OLC has motivated other researchers to find an alternative approach. Pevzner[16] attempted to solve this problem by introducing an Eulerian path approach using a de Bruijn graph. In this approach, elements are not organized around reads but around words of k-nucleotides, called k-mers. This approach transposes the difficulties of the OLC problem in dealing with sequence assembly for very short reads into an Eulerian path problem by using a de Bruijn graph. Some of the well-known assemblers based on the Eulerian path approach include EULER[16], SHARGS[17], Velvet[15], and ALLPATHS[18].

The performance of a DNA assembler is commonly evaluated by measures such as the number of contigs, the N50 size and the maximum contig length. The N50 size is the size of the smallest contig such that the total length of all of the contigs

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

greater than this size is at least one half of the total genome size. In addition, the final contigs produced by a DNA assembler also should be evaluated by aligning the contigs to a reference sequence to view the number of mismatch contigs and the genome coverage. An assembler is considered to have a high performance if it yields fewer contigs with a longer maximum contig length and a longer N50 size than those of other assemblers. Moreover, it also should produce fewer mismatch contigs (a high accuracy of the contigs) and have reasonable genome coverage. Therefore, there is a trade-off between the contig size and the accuracy.

Velvet, one of the most popular assemblers based on Eulerian paths, has successfully showed the effectiveness of the Eulerian path approach in assembling very short reads by yielding longer contigs in terms of the N50 size and the maximum contig lengths [15)]. However, Edena, an assembler based on OLC, could also show similar performance as shown by Velvet in assembling Illumina's very short reads. Velvet and Edena use different approaches, including different data representations, different strategies to simplify the graphs, and different algorithms to find the shortest paths as a solution. Intuitively, applying different approaches will obtain contigs that may cover different regions in the genome. The author of Edena also indicated this possibility by showing some simple results [11)]. However, no researchers have deeply elaborated upon it.

In this research, we have elaborated upon previous methods and proposed a new assembly technique by combining the OLC and Eulerian path in a hierarchical process. Contigs yielded by two assemblers with different approaches will be treated as reads and will be assembled again. This research was conducted to assess the feasibility of yielding longer accurate contigs in *de novo* sequence assembly for very short reads from next generation sequencers, particularly for Illumina's very short reads.

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 [19)]. Suppose we have a DNA sequence from an unknown source and the sequence is $A = a_1, a_2, \ldots, a_L$. Shotgun sequencing of sequence A produces a set of reads (or fragments) $F = f_1, f_2, \ldots, f_R$ that are sequences over the alphabet $\Sigma = \{A, C, G, T\}$. To reconstruct the sequence A from the set of its fragments $F = f_1, f_2, \ldots, f_R$, 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 [20)]. Let $s_1 = a_1, \ldots, a_r$ and $s_2 = b_1, \ldots, b_r$ be strings over some finite alphabet Σ. 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 $i \leq j \leq r$.

The SCS is a simple model for formulating the problem of genome assembly. It 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 [21)]. However, in the real problem, the assembly process becomes more complicated because of the presence of sequencing errors, polymorphisms and repeats that are not considered by the SCS. Therefore, because of the limitation of SCS, researchers formulate the *de novo* DNA assembly problem as a graph traversal problem. In this formulation, there are two main approaches: 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) $s_1, s_2, \ldots, s_n$ and insert edges of length overlap (s_i, s_j) between the vertices s_i and s_j. For this purpose, Medvedev et al. [21)] define overlap in a general form, as follows: let v and w be two strings over the alphabet $\Sigma = \{A, C, G, T\}$. 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. Here, $|z|$ 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. This problem is a Hamiltonian path problem, which is known to be NP-hard. Moreover, this overlap graph formulation is in fact not suitable for finding the single path that represents the shortest superstring [22)]. Furthermore, in the layout stage, the set of contigs is connected to yield supercontigs. For this purpose, we need information on the mate pair lengths to estimate the distance between the two contigs that are to be connected. The final step in the OLC strategy is consensus. The goal of this step is to determine the DNA sequence by aligning all of the reads that cover 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[16] 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[21]. Let $S = s_1, s_2, \ldots, s_n$ be a set of strings over an alphabet $\Sigma = \{A, C, G, T\}$ and let $G = B_k(S)$ be the de Bruijn graph of S for some k. The string s_i corresponds to walks in $B_k(S)$ via the function $w(s) = s[1 \ldots k] \rightarrow [2 \ldots k+1] \rightarrow \ldots \rightarrow s[|s|-k+1, |s|]$. A walk is called a superwalk of G if, for all i, it contains $w(s_i)$ as a subwalk. Thus, a superwalk represents a valid assembly of the reads into a genome. Formally, given a set string S, as defined above, and a 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

There are three important problems in DNA assembly: unknown orientations, the presence of repeats, and the existence of sequencing errors. In this study, we investigated how these problems are handled by each approach. We used two assemblers (Edena and Velvet) as a case. We choose Edena and Velvet because they are among the best DNA assemblers in assembling very short reads and represent each of the two major approaches.

Edena adopts a bi-directed overlap graph to deal with the unknown orientation problem. In this graph, each node corresponds to a double stranded read (a read and its reverse complement), and each overlap (edge) has an orientation at both endpoints[21]. On the other hand, Velvet adopts a bi-directed de Bruijn graph, to overcome the unknown orientation problem. In a bi-directed de Bruijn graph, the nodes of the graph will be all of the possible (k-1)-mers and their complement.

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. However, Edena has successfully demonstrated that the OLC approach can also yield significant contigs by employing a suffix array to perform the overlapping phase[11]. 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 graph construction. 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, tips and bubbles.

Thus, because different data representations and different strategies for reducing sequencing errors and 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, then these two assemblers may be complementary to each other and yield longer contigs[11].

3. Materials and Methods

3.1 Dataset and Computer Resources

We examined our approach by using three real datasets and four simulated datasets. For real datasets, we used sequences of *Staphylococcus aureus* strain MW2[11], *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 consists of 3.86 million 35-bp reads. The raw coverage depth is 48x. The *Helicobacter pullorum* dataset consists of 4.53 million 36-bp reads. The *Bacillus anthracis* BA104 dataset consists of 7.63 million 50-bp reads.

Moreover, we also 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[23]; the sizes of these datasets were 2.91 million bp, 6.52 million bp, 2.01 million bp, and 4.94 million bp, respectively. We generated three groups of datasets from each organism. The first group was set as error-free reads with coverage depths 10x, 20x, 30x, 40x, and 50x. The length of the reads is 36 bp. These reads can be yielded by choosing the Exact Error Model in MetaSim. The second group was set as 36-bp reads that contain errors. Errors were simulated based on Illumina's sequencing technology [22)] and were uniformly distributed over the reads. The second group of datasets was generated with a coverage depth of 10x, 20x, 30x, 40x, and 50x. The third group was simulated as datasets that contain errors based on the Illmunina Error Model normally distributed over the reads, for a coverage depth of 48x. We only considered substitution errors as the most common class of error for current short read sequencing technologies [14)].

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 GHz 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 Proposed Method

In this research, we proposed to assemble very short reads hierarchically by combining the OLC and the Eulerian path methods. We can consider this method to be a combined approach. In our combined approach, contigs yielded by the two approaches can be treated as reads and can be assembled again to yield longer contigs than the previous contigs. We formulate our combined approach in a general form, as follows: Let $E = e_1, \ldots, e_n$ be a set of contigs that are produced by assemblers based on the Eulerian path approach and let $O = o_1, \ldots, o_n$ 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. From a general viewpoint, we can see the goal of the problem to be finding an ordering of the strings that maximizes the amount of overlap between consecutive strings [20)], as follows: Let $e_1 = a_1 \ldots a_r$ and $o_1 = b_1 \ldots b_r$ be strings. We define: $(e_1, o_1) = max\{k \geq 0 | a_{r-k+i} = b_i, 1 \leq i \leq k\}$. We now introduce our combined approach (**Fig. 1**), which successfully yields longer contigs. This approach combines the OLC and the Eulerian methods and is divided into three phases. We applied three assemblers as part of a combined approach, including Edena (based on OLC), Velvet (based on the Eulerian path method), and Minimus (based on OLC), as initially suggested by Hernandez [11)].

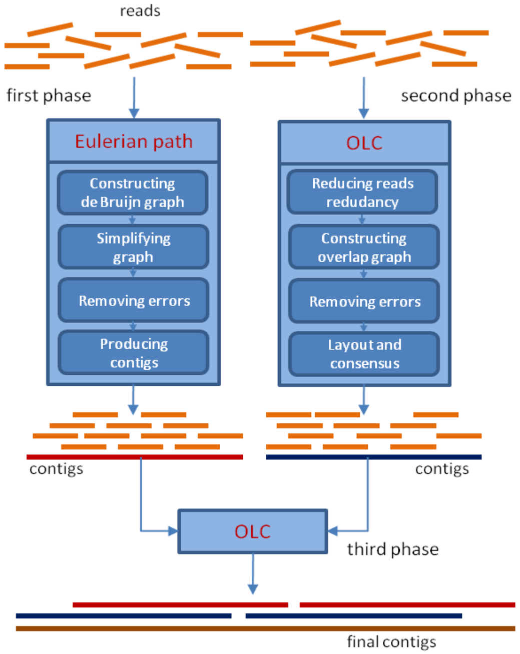


Fig. 1 The flowchart of our combined approach. First, Illumina reads were assembled by Velvet and Edena separately. Next, we assembled contigs produced by Velvet and Edena using the Minimus assembler to produce longer contigs than previous method.

In the first phase, we assembled very short reads with Velvet. Velvet is started by hashing reads according to a predefined length to construct a bi-directed de Bruijn graph. These graphs are simplified by merging chains of nodes into single nodes. Furthermore, to remove sequencing errors from the existing graph, some topologies such as erroneous edges, tips and bubbles are removed. Final contigs can be yielded by finding a superwalk, as described in Section 2.2. In the experiments with Velvet, we should choose the optimal value of k for getting the best results.

In the second phase, the same datasets were assembled by Edena. First, redundant reads are reduced by indexing reads in a prefix tree. Next, in the overlap step, overlaps are detected according to the criteria described in Section 2.1. Then, a bi-directed overlap graph is constructed. After constructing the graph, sequencing errors are eliminated by removing transitive and spurious edges and resolving bubbles. Finally, all significant contigs are generated in the layout and consensus step. In all experimental scenarios, Edena was run in strict mode (do not assemble ambiguities) and was parameterized to consider overlaps displaying a minimum length of 22 bases.

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 yielded by Velvet. The merging contigs were then assembled using Minimus. Minimus is a DNA assembler based on OLC, which can assemble fragments with different lengths. Therefore, Minimus is appropriate to assemble contigs yielded by Edena and Velvet. Similar to other assemblers based on OLC, Minimus starts with detecting overlap by computing all pairwise alignments between the reads. Next, an overlap graph is constructed and several simplification and error removal procedures based on algorithms developed by Myers[24] are performed, such as the removal of containment edges, transitive reduction, and unique-join collapsing[12]. In the final step, consensus, the final contigs are yielded by performing the full multiple alignments of the aligned reads. This approach combines the Eulerian path and the OLC methods, which are represented by the Velvet and Edena assemblers, respectively.

3.3 Performance Evaluation

Before evaluating the overall performance of the DNA assemblers, the ability of each approach to reduce sequencing errors was evaluated. To confirm the effectiveness of each error removal procedure, the assembly processes were performed on error-free reads and on reads containing Illumina errors. Both reads were generated by MetaSim in a variety of coverage depths. Next, the results were evaluated by comparing the value of 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 the largest to the smallest and by determining the minimum set of contigs whose sizes total 50% of the entire genome.

Furthermore, the overall performance of a DNA assembler in producing contigs was evaluated by measures such as the number of contigs, the N50 size and the maximum contig length. In addition, to measure the accuracy of the DNA assemblers, the final contigs yielded by assemblers were evaluated by aligning the contigs to the reference sequence to view the number of mismatch contigs and the genome coverage.

4. Results and Discussion

4.1 Testing Error Removal on Simulated Data

To evaluate the effectiveness of the error removal procedure of each approach, we compared the results of assembling error free reads or ideal reads and reads that contain Illumina errors as presented by Zerbino et al. in their paper[15]. We used the first group dataset and the second group dataset, as described in Section 3.1. The first group represents error-free reads or ideal reads and the second group dataset represents reads containing Illumina errors that are uniformly distributed over the reads. For this test, we used two sequences: *Acetobacter pasteurianus* IFO 3283-01, and *Streptococcus suis* P17. Both datasets were assembled using Velvet (Eulerian path), Edena (OLC), and our combined approach. Velvet and the combined approach were executed with k = 21 or used 21-mer words to construct a de Bruijn graph.

Figure 2 shows that for low coverage depth (from 10x to 30x), the N50 size of the contigs produced by all of the approaches from reads containing Illumina errors (color curve) is lower than the N50 size of the contigs yielded from error free reads (black curve). However, when the coverage is sufficient (from 30x to 50x), our combined approach could obtain almost the same value of the N50 size for both tests, using error free reads and reads that contain errors. This information indicates that our combined approach could reduce almost all of the sequencing errors within the reads, provided that the coverage depth of the reads is sufficient. On the other hand, the N50 size of Velvet with Illumina error reads is always lower than the size shown using error free reads, although the difference between the N50 size values is not significant. Moreover, for a low coverage depth (<40x), the error removal procedure of Edena is not very effective. The reason may be because, in a low coverage depth, the possibility of generating overlapping reads is decreased. Therefore, the overlap graph constructed by Edena is not repre-

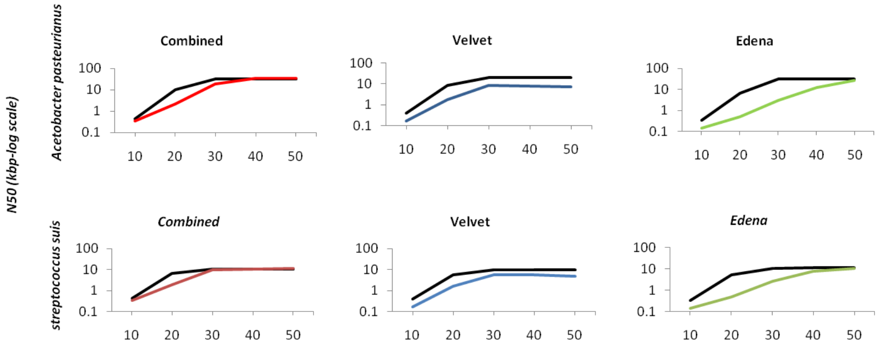


Fig. 2 The genome of *Acetobacter pasteurianus* and *Streptococcus suis* were used to generate 36-bp read sets of varying coverage depths (from 10x to 50x). We measured the N50 size to show the performance of each approach. The results of each approach regarding the assembly of reads that contain Illumina errors are shown by the color curves; red, blue, and green indicate our combined approach, Velvet, and Edena, respectively. Moreover, the black curve represents the results of each approach regarding assembling error free reads (ideal reads).

sentative. However, the overlap region can be found significantly over the reads after the coverage depth is sufficient (from 40x to 50x). The representative overlap graph can be constructed and the error removal procedure can be executed effectively.

In addition, Fig. 2 shows that the N50 increases significantly in a low coverage depth, but this tendency is changed when the coverage depth of the reads is sufficient. The N50 size stops increasing because of the presence of natural repetitions in the genome [15].

4.2 Performance of the Approach with the Simulated Dataset

The performance of a DNA assembler was evaluated by measures including the number of contigs, the N50 size and the maximum contig length. The N50 size is an important measure for demonstrating the ability of a DNA assembler to yield significant contigs. A comparison of the N50 size for each approach is presented here at a variety of coverage depths. In this experiment, we used k=21 for executing Velvet and our combined approach. For this test, we used two sequences: *Acetobacter pasteurianus* IFO 3283-01, and *Streptococcus suis* P17. The results (**Fig. 3**) show that our combined approach always yields an N50 size of greater value than the sizes yielded by Velvet and Edena in all of the coverage depths.

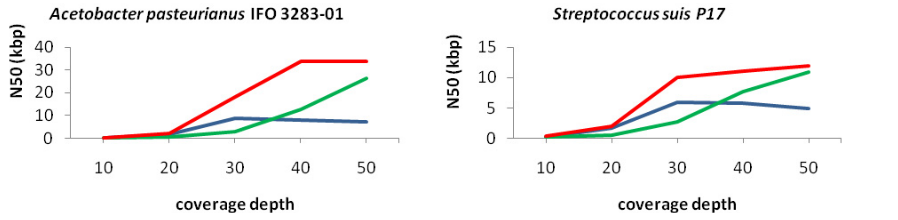


Fig. 3 The 36-bp simulated reads generated from *Acetobacter pasteurianus* and *Streptococcus suis* were assembled using our combined approach (red curve), Velvet (blue curve), and Edena (green curve). When assembling with Velvet and our combined approach, we chose **k=21**. We used one important measure, the N50 size, to evaluate the capability of each approach to yield significant contigs in varying coverage depths (from 10x to 50x).

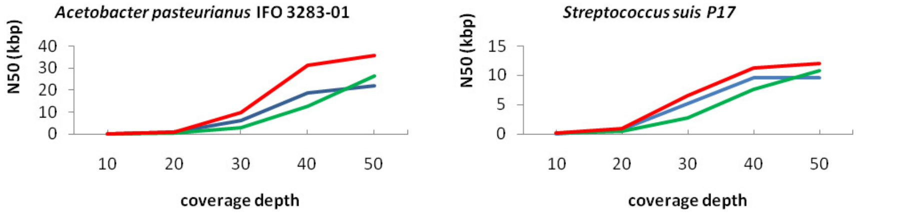


Fig. 4 The 36-bp simulated reads generated from *Acetobacter pasteurianus* and *Streptococcus suis* were assembled using our combined approach (red curve), Velvet (blue curve), and Edena (green curve). When assembling with Velvet and our combined approach, we chose **k=23**. We used one important measure, the N50 size, to evaluate the capability of each approach to yield significant contigs at varying coverage depths (from 10x to 50x).

Because the results of Velvet depend on the value of k, we also executed Velvet and our combined approach using k=23. Both values (k=21 or k=23) are recommended by Zerbino [15] when we assemble 36-bp reads. **Figure 4** shows that, although the results of Velvet are improved, our combined approach still has better performance than Velvet and Edena at all coverage depths. It is indicated that the contigs produced by Velvet and Edena cover different regions in the genome and contain overlapping regions between one another.

The tendencies of each curve shown in Figs. 3 and 4 indicate that the high-

Table 1 Comparison of Velvet, Edena, and our combined approach with simulated reads that contain Illumina errors uniformly distributed over the reads.

Genome	Assembler	Number of contigs	Maximum length of contigs (kbp)	N50 (kbp)	Total length of contigs (Mbp)
Acetobacter pasteurianus	Velvet (k=21)	710	33.33	7.9	2.76
	Edena	518	54.5	12.7	2.74
	Combined (k=21)	**289**	**234.8**	**33.7**	2.75
Rhodobacter erythropolis	Velvet (k=23)	1,422	59.5	10.7	6.45
	Edena	1,595	43.0	8.7	6.44
	Combined (k=23)	**1,043**	**74.9**	**15.7**	6.45
Streptococcus suis	Velvet (k=23)	440	**33.4**	9.6	1.96
	Edena	404	24.3	7.7	1.95
	Combined (k=23)	**368**	**33.4**	**11.2**	1.95
Escherichia coli	Velvet (k=21)	1,810	28.1	5.3	4.71
	Edena	1,237	47.5	10.0	4.08
	Combined (k=21)	**763**	**70.5**	**17.9**	4.66

Table 2 Comparison of Velvet, Edena, and our combined approach with simulated reads that contain Illumina errors normally distributed over the reads.

Genome	Assembler	Number of contigs	Maximum length of contigs (kbp)	N50 (kbp)	Total length of contigs (Mbp)
Acetobacter pasteurianus	Velvet (k=23)	376	172.5	19.7	2.76
	Edena	393	84.4	25.6	2.74
	Combined (k=23)	**280**	**249.1**	**32.4**	2.75
Rhodobacter erythropolis	Velvet (k=25)	984	74.9	16.3	6.45
	Edena	1,375	51.3	11.4	6.45
	Combined (k=25)	**777**	**89.7**	**21.7**	6.45
Streptococcus suis	Velvet (k=25)	405	33.4	11.0	1.96
	Edena	450	26.8	10.0	1.95
	Combined (k=25)	**342**	**33.5**	**12.5**	1.96
Escherichia coli	Velvet (k=25)	897	**67.1**	15.7	4.83
	Edena	1,013	64.0	15.2	4.80
	Combined (k=25)	**713**	**67.1**	**19.9**	4.81

est performance of our combined approach is obtained with a coverage depth of reads from 40x to 50x. Therefore, we performed experiments using four simulated datasets that have coverage depths of 40x and 48x. We used the second group and third group datasets described in Section 3.1. The second group dataset represents 36-bp reads containing Illumina errors uniformly distributed over the reads. This dataset was generated with a coverage depth of 40x. Moreover, the third group dataset represents 36-bp, which contains Illumina errors that are normally distributed over the reads with a coverage depth of 48x. All datasets were generated by MetaSim from four organisms, including the sequences of *Acetobacter pasteurianus* IFO 3283-01, *Rhodobacter erythropolis* PR4, *Streptococcus suis* P17, and *Escherichia coli* 536. The aim of these two experiments is to evaluate the ability of our combined approach to yield fewer contigs than Velvet and Edena under similar conditions but also to yield higher N50 size values and maximum contig lengths.

The results show that our combined approach always yields fewer contigs than Velvet and Edena. Moreover, the maximum length of the contigs and the N50 size yielded by our combined approach are increased, except for in the *Streptococcus suis* dataset (**Table 1**). The most significant contigs yielded by our combined approach are obtained using the *Acetobacter pasteurianus* dataset. The maximum length of the contigs is 200 kbp and 180 kbp longer than Velvet and Edena, respectively. The N50 size of the contigs produced by our combined approach is also longer than the sizes of Velvet and Edena.

The same tendency is also indicated from the results of the assembly of simulated reads, which contain Illumina errors that are normally distributed over the reads (**Table 2**). The coverage depth of this dataset is 48x. The high coverage depth significantly affects the performance of DNA assemblers. All of the results of assemblers presented in Table 2 are higher than those presented in Table 1.

4.3 Performance of the Approach with Real Datasets

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 and our combined approach were used with k-mer values of 23 for the *Staphylococcus aureus* strain MW2 and *Helicobacter pullorum* NCTC 12824, k=31, for *Bacillus anthracis* BA104. 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 show that using our combined approach with the *Staphylococcus aureus* strain MW2 dataset yielded 890 contigs fewer than Velvet (1,152) and Edena (1,175). Moreover, the maximum length and the N50 size of the contigs produced by our combined approach are all longer than

Table 3 Comparison of Velvet, Edena, and our combined approach with a real Illumina dataset.

Genome	Assembler	Number of contigs	Maximum length of contigs (kbp)	N50 (kbp)	Total length of contigs (Mbp)
Staphylococcus aureus	Velvet (k=23)	1,152	22.89	5.30	2.78
	Edena	1,175	22.89	5.46	2.76
	Combined (k=23)	**890**	**32.73**	**7.40**	2.77
Helicobacter pullorum	Velvet (k=23)	3,981	4.38	0.58	1.76
	Edena	4,300	4.11	0.35	1.26
	Combined (k=23)	**2,570**	**9.20**	**0.86**	1.66
Bacillus anthracis	Velvet (k=31)	3,436	58.11	4.70	5.05
	Edena	2,203	23.14	5.06	5.03
	Combined (k=31)	**1,235**	**71.13**	**11.80**	5.02

Table 4 The results of aligning contigs to the reference sequence of Velvet, Edena, and our combined approach.

Genome	Assembler	Number of contigs	Number of mismatch contigs	Genome coverage
Staphylococcus aureus	Velvet (k=23)	1,152	16	97%
	Edena	1,175	46	96%
	Combined (k=23)	**890**	**17**	97%
Acetobacter pasteurianus	Velvet (k=21)	710	16	94%
	Edena	518	29	94%
	Combined (k=21)	**289**	**10**	94%
Rhodobacter erythropolis	Velvet (k=23)	1,422	**14**	98%
	Edena	1,595	44	98%
	Combined (k=23)	**1,043**	**14**	98%
Escherichia coli	Velvet (k=21)	1,810	40	97%
	Edena	1,237	51	97%
	Combined (k=21)	**763**	**31**	97%

the corresponding numbers yielded by Velvet and Edena. The N50 value and the maximum contig length increased to 2 kbp and 10 kbp, respectively (**Table 3**). The maximum contig length is 32.73 kbp, and the N50 value is 7.4 kbp.

A similar tendency is also indicated through the use of the *Helicobacter pullorum* dataset. Our combined approach yields 2,570 contigs fewer than Velvet (3,981) and Edena (4,300), and the N50 value and maximum length of contigs increased to 300 bp and 5 kbp, respectively (Table 3).

Moreover, the increased maximum length and the N50 were large when the *Bacillus anthracis* BA104 dataset was used. The N50 value and the maximum length of the contigs are 6 kbp and 50 kbp longer, respectively, than Velvet and Edena alone.

4.4 Aligning Contigs to the Reference Sequence

Evaluating the performance of a DNA assembler using measures such as the number of contigs, the N50 size, and the maximum contig length, as presented above, is not sufficient. The accuracy of contigs yielded by a DNA assembler should be evaluated by aligning contigs to the reference sequence to find the number of mismatch contigs and to calculate the genome coverage, thus indicating what ratio of the region of the genome can be covered by all of the contigs. We used the reference sequences of *Staphylococcus aureus* MW2, *Acetobacter pasteurianus* IFO 3283-01, *Rhodobacter erythropolis* PR4, and *Escherichia coli* 536 from the NCBI database. Contigs were aligned to their reference sequence using the nucmer module from the Mummer sequence alignment package[25].

To be considered valid, a contig must be aligned along its whole length with a base similarity of at least 98%. Only contigs larger than, or equal to, 100 bases were considered because short contigs are more likely to contain base errors at the ends[11]. We used the contig results of *Staphylococcus aureus* MW2 presented in Table 3 and those of *Acetobacter pasteurianus* IFO 3283-01, *Rhodobacter erythropolis* PR4, and *Escherichia coli* 536 presented in Table 1.

Table 4 shows that almost all of the contigs yielded by our combined approach have fewer mismatch contigs than those yielded by Velvet and Edena. As indicated from the genome coverage, the size of the mismatch contigs is not significant because the genome coverage of the assemblers is still in a high range, from 94% to 98%.

4.5 Comparing Our Approach with the Recent Assembler SOAPdenovo

We considered comparing our combined approach with the recent DNA assembler SOAPdenovo[26], which was developed based on the Eulerian path approach using a de Bruijn graph. This assembler is assigned not only for yielding contigs but also for producing scaffolds. However, in this comparison, we only consider the function of producing contigs.

We executed the assemblers at several coverage depths from 10x to 50x using

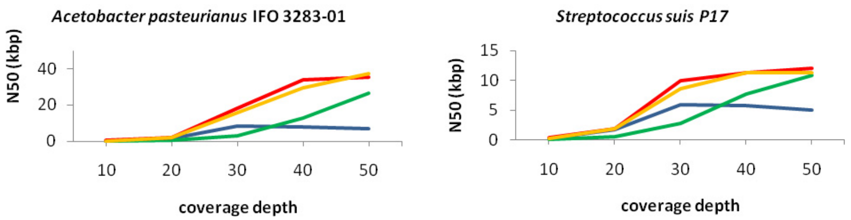


Fig. 5 The 36-bp simulated reads generated from *Acetobacter pasteurianus* and *Streptococcus suis* were assembled using our combined approach (red curve), SOAPdenovo (yellow curve), Velvet (blue curve), and Edena (green curve). We used one important measure, the N50 size, to evaluate the capability of each approach to yield contigs at varying coverage depths (from 10x to 50x).

Acetobacter pasteurianus and *Streptococcus suis* datasets, which contain sequencing errors uniformly distributed over the reads. The program was run at several values of k, from k=19 to k=25, to find the optimal value of k. The results with the optimal value of k for each coverage depth are presented in **Fig. 5**. In our case, the optimal value of k could be obtained by choosing either k = 21 or k = 23, as mentioned by Zerbino[15]. The results show that our combined approach could be comparable with SOAPdenovo. Moreover, both our combined approach and SOAPdenovo have better performance than Velvet and Edena.

In addition, we also evaluated each approach with the simulated sequences of *Acetobacter pasteurianus* IFO 3283-01, *Rhodobacter erythropolis* PR4, *Streptococcus suis* P17, and *Escherichia coli* 536 containing Illumina sequencing errors with a uniform distribution. All of them have a coverage depth of 40x.

From the results shown in **Table 5**, we can see that the performance of our combined approach in producing significant contigs is comparable to the performance of SOAPdenovo. In some results, as shown by using the *Streptococcus suis* and the *Escherichia coli* datasets, the performance of our combined approach is slightly better than the performance of SOAPdenovo. The most significant contigs yielded by our combined approach are obtained using the *Acetobacter pasteurianus* dataset with k=21. The maximum length of the contigs and the N50 are 42 kbp and 10 kbp, respectively, which is longer than those for SOAPdenovo. The significant results of our combined approach are also obtained by using the Illumina real dataset (**Table 6**). However, with the *Rhodobacter erythropolis* dataset, the results of SOAPdenovo are slightly higher than the results of our combined approach (Table 5).

Table 5 Comparison of SOAPdenovo and our combined approach with simulated reads that contain Illumina errors uniformly distributed over the reads.

Genome	Assembler	Number of contigs	Maximum length of contigs (kbp)	N50 (kbp)	Total length of contigs (Mbp)
Acetobacter pasteurianus	SOAPdenovo (k=21) Combined (k=21)	319 **289**	129.06 **234.8**	23.30 **33.7**	2.75 2.75
Rhodobacter erythropolis	SOAPdenovo (k=23) Combined (k=23)	**969** 1,043	**80.14** 74.90	**16.31** 15.73	6.44 6.45
Streptococcus suis	SOAPdenovo (k=23) Combined (k=23)	396 **368**	**33.45** **33.43**	**11.21** **11.22**	1.96 1.95
Escherichia coli	SOAPdenovo (k=23) Combined (k=21)	850 **763**	67.09 **70.54**	**18.91** 17.89	4.81 4.80

Table 6 Comparison of SOAPdenovo and our combined approach with the real Illumina dataset.

Genome	Assembler	Number of contigs	Maximum length of contigs (kbp)	N50 (kbp)	Total length of contigs (Mbp)
Staphylococcus aureus	SOAPdenovo (k=21) Combined (k=21)	912 **772**	30.86 **32.73**	7.90 **8.89**	2.77 2.77
Helicobacter pullorum	SOAPdenovo (k=21) Combined (k=21)	2,567 **2,369**	5.37 **7.69**	**1.07** 0.95	1.79 1.68
Bacillus anthracis	SOAPdenovo (k=31) Combined (k=31)	2,749 **1,235**	58.14 **71.13**	6.52 **11.80**	5.03 5.02

To measure the real performance of our combined approach compared to the SOAPdenovo, we assembled some real Illumina datasets, as previously used in Table 3, such as for the *Staphylococcus aureus* strain MW2, *Helicobacter pullorum* NCTC 12824 and *Bacillus anthracis* BA104 from the NCBI Short Read Archive. SOAPdenovo and our combined approach were used with k-mer values of 21 for *Staphylococcus aureus* strain MW2 and *Helicobacter pullorum* NCTC 12824, and k= 31 for *Bacillus anthracis* BA104.

The results (Table 6) show that our combined approach always performs better than SOAPdenovo in producing significant contigs, except for the N50 size of the contigs that are yielded when assembling *Helicobacter pullorum* with k = 21,

Table 7 The results of aligning contigs to the reference sequence of SOAPdenovo and our combined approach.

Genome	Assembler	Number of contigs	Number of mismatch contigs	Genome coverage
Staphylococcus aureus	SOAPdenovo (k=21)	912	32	97%
	Combined (k=21)	**772**	**17**	97%
Acetobacter pasteurianus	SOAPdenovo (k=21)	349	14	94%
	Combined (k=21)	**289**	**10**	94%
Rhodobacter erythropolis	SOAPdenovo (k=23)	**969**	19	98%
	Combined (k=23)	1,043	**14**	98%
Escherichia coli	SOAPdenovo (k=23)	850	38	97%
	Combined (k=21)	**763**	**31**	97%

although the difference is not significant (120 bp). The most significant contigs yielded by our combined approach are obtained using the *Bacillus anthracis* dataset. The maximum length of the contigs and the N50 size are 13 kbp and 5 kbp, respectively, which are larger than those for SOAPdenovo.

All of the results that are presented above (Fig. 5, Tables 5, 6, **Table 7**) show the performance of DNA assemblers in yielding significant contigs. As mentioned before in Section 4.4, evaluating the performance of a DNA assembler from measures such as the number of contigs, the N50 size, and the maximum contig lengths, as presented above, is not sufficient. We require measuring the accuracy by aligning contigs to the reference sequence. We used the reference sequence as previously used in Table 4. Contigs were aligned to their reference sequence using the nucmer module from the Mummer [25]. To be considered valid, a contig must be aligned along its whole length with a base similarity of at least 98%. Only contigs larger than, or equal to, 100 bases are considered.

The results show that the accuracy of our combined approach is higher than SOAPdenovo for all datasets, when using the optimum value of k for each dataset. Our combined approach produced fewer mismatch contigs than the number of mismatch contigs found by SOAPdenovo (Table 7) and could maintain genome coverage in a high range, from 94% to 98%.

4.6 Assembling a Dataset with Various Values of k

As presented in Table 2–7, the effectiveness of the Velvet and SOAPdenovo assembler in producing significant contigs depends on the value of k. The value of k becomes a critical point in constructing a de Bruijn graph. Smaller k-mers increase the connectivity of the graph. Given this condition, the sensitivity will increase. 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 is highly important [15].

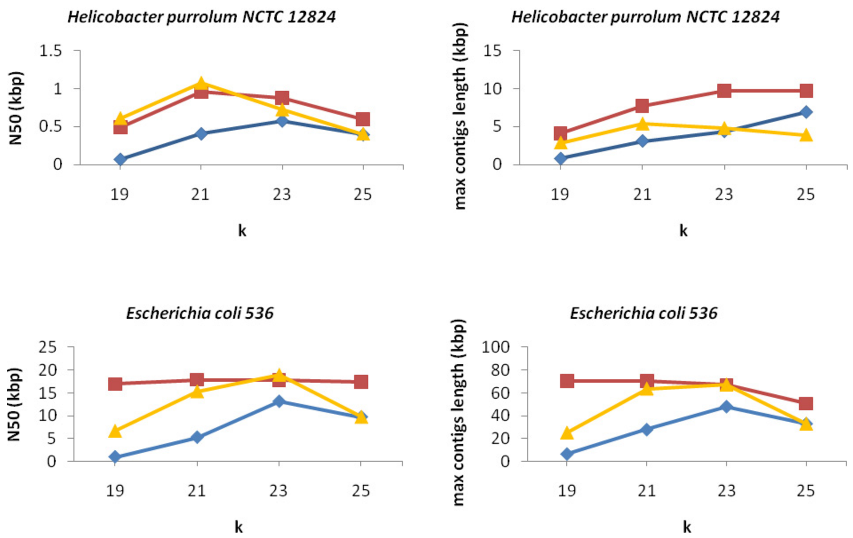


Fig. 6 The real datasets of *Helicobacter pullorum* NCTC 182824 and the 36-bp simulated reads generated from *Escherichia coli* 536 were assembled using our combined approach (red curve), SOAPdenovo (yellow curve) and Velvet (blue curve) for various values of k. The N50 size and the maximum contig lengths of both assemblers were compared. The results show that our combined approach has a better performance than Velvet and SOAPdenovo for any value of k.

To study the effects of the value of k, we executed Velvet, SOAPdenovo, and our combined approach 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 was to evaluate the performance of the approach with a variety of k-values. However, it is important to note that k must be an odd value because in Velvet each k-mer is recorded simultaneously with its reverse complement.

Figure 6 shows that our combined approach obtained higher values regarding the N50 size and the maximum contig lengths than Velvet and SOAPdenovo for almost all values of k, between k=19 to k=25. Except for *Helicobacter pullorum*, the N50 size of SOAPdenovo is slightly higher than our combined approach with k=19 and k=21. There is a tendency for our combined approach to perform better in assembling reads by choosing any odd value of k (from 19 to 25) for assembling 36-bp reads in comparison with Velvet and SOAPdenovo.

4.7 Discussion

Todays DNA sequencing equipment, commonly referred to as next-generation sequencers[27], produce shorter reads, such as in the 400 bp range (from 454 machines), the 100 bp range (from the Solexa and SOLiD machine), or less[27], from 35 bp to 50 bp (from the Solexa). Assembly of shorter reads, particularly very short reads, requires higher coverage, to meet the minimum overlap criteria. High coverage increases complexity. Thus, assembling shorter reads, especially very short reads, is still an important issue.

Many assemblers have been developed. Most of them proposed improvements at the technical level, based on the two main approaches, overlap-layout-consensus (OLC) and Eulerian paths using de Bruijn Graphs. As previously mentioned, the graph representations are much more complicated with the presence of sequencing errors, polymorphisms, and repeats. Therefore, the problem is not just how to find the Hamiltonian and the Eulerian path in the OLC and Eulerian path approach, respectively. More procedures are required to correct or remove sequencing errors and polymorphisms as well as to simplify the graph before finding paths as solutions. Many assemblers were developed by proposing new techniques or a combination of techniques at this level. Commonly, these techniques are adopted from other assemblers.

We can consider the features of SOAPdenovo, which adopts some techniques from Euler and Velvet. In SOAPdenovo, there is a preprocessing step for correcting sequencing errors by using pre-set thresholds for k-mer frequencies[27]. This technique is similar to the technique of Euler. Moreover, to remove polymorphisms, SOAPdenovo adopts Velvets technique by removing bubbles. SOAPdenovo also reduces complexity in the graph by removing transitive edges as similar in Velvet. Therefore, SOAPdenovo could outperform Velvet in terms of the N50 size and the maximum contig lengths. Unlike Velvet, which only implements sequencing errors and bubble removal procedures, SOAPdenovo employs both the sequencing error corrections and the error or bubble removal. This structure is one of the reasons why SOAPdenovo could produce a higher N50 size than Velvet.

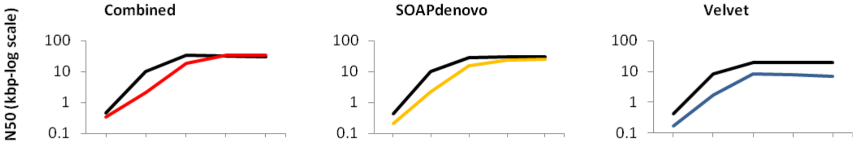


Fig. 7 The genome of *Acetobacter pasteurianus* was used to generate 36-bp read sets of varying coverage depths (from 10x to 50x). We measured the N50 size to show the performance of each approach. The results of each approach regarding the assembly of reads that contain Illumina errors are shown by the color curves; red, yellow, and blue indicate our combined approach, SOAPdenovo, and Velvet, respectively. Moreover, the black curve represents the results of each approach regarding assembling error free reads (ideal reads).

Although our combined approach does not apply a sequencing error correcting procedure, it employs a hierarchical process when dealing with sequencing errors and polymorphisms. The sequencing errors and polymorphisms are removed in three phases. In the first phase, the sequencing errors and polymorphisms are removed based on the Velvet error removal procedure by removing bubbles and tips from the de Bruijn graph that was constructed. In the second phase, the sequencing errors are eliminated by removing transitive and spurious edges and resolving bubbles according to Edena. Finally, the error removal procedures based on the algorithms developed by Myers[24] were performed using Minimus, such as the removal of containment edges and transitive reduction. **Figure 7** shows that by employing a hierarchical sequencing error removal procedure, our combined approach could obtain similar results to SOAPdenovo. At a high coverage depth (from 40x to 50x), our combined approach could perform slightly better than SOAPdenovo.

In addition, our combined approach introduces the combining of two approaches: OLC and Eulerian path. This strategy was considered based on the intuitive view that contigs produced by two assemblers employ two different approaches that cover different regions in the genome. All of the results show the effectiveness of our combined approach in yielding more significant and accurate contigs than assemblers that employ only one approach, such as SOAPdenovo,

Table 8 Comparison of our combined approach, SOAPdenovo and some other combined assemblers with *Staphylococcus aureus* dataset.

Genome	Assembler	Number of contigs	Maximum length of contigs (kbp)	N50 (kbp)	Total length of contigs (Mbp)
Staphylococcus aureus	SOAPdenovo	1035	26.97	6.28	2.78
	SOAPdenovo + Velvet	985	26.97	6.43	2.78
	SOAPdenovo + Edena	**874**	**32.73**	**7.82**	2.78
	Combined approach	**890**	**32.73**	**7.40**	2.78

Velvet, and Edena.

It is also interesting to notice that combining two assemblers that employ the same approach may not be effective for generating significant contigs. **Table 8** supports this statement. We executed SOAPdenovo and Velvet for assembling *Staphylococcus aureus*. Then, contigs produced by these two assemblers are assembled again using Minimus. Both SOAPdenovo and Velvet are developed based on a Eulerian path approach using a de Bruijn graph. As presented in Table 8, the results of SOAPdenovo and Velvet could not outperform the results of our combined approach, and are almost similar to those of SOAPdenovo. However, by combining SOAPdenovo and Edena, which employ different approaches (OLC and Eulerian paths), we could obtain results similar to the results of our combined approach, which uses Velvet and Edena. Therefore, it is expected that the idea of combining two approaches can be generalized by formulating a solution at a conceptual level, such as in a graph representation.

5. Conclusions

An effective approach for DNA assembly of Illuminas very short reads has been performed to yield longer contigs by involving the combination of two main approaches, the OLC and the Eulerian path methods.

It was observed that our combined approach can improve the ability of assemblers regarding sequencing error removal, particularly at coverage depths ranging from 30x to 50x. Applying our combined approach to assemble real datasets at a sufficient coverage depth can also yield contigs longer than those of SOAPdenovo, Velvet and Edena, in terms of the maximum contig lengths, and higher than those of SOAPdenovo, Velvet and Edena, with respect to the N50 size. The accuracy of the contigs produced by our approach also increased, as indicated by a decreasing occurrence of mismatch contigs compared with SOAPdenovo, Velvet and Edena. Our combined approach can also maintain genome coverage, as indicated by some of the results. In addition, unlike Velvet, which requires finding an optimal value of k to obtain longer contigs, our results indicated that longer contigs can be determined by applying our combined approach with any odd value of k in the range of 19 and 25 for 36-bp reads.

Considering its advantages, our combined approach can be considered as a more robust and effective approach in *de novo* DNA assembly that produces significant and accurate contigs from very short reads generated by next-generation sequencers.

Availability A script of our combined approach and a code used in the analysis are freely available at http://www.bi.cs.titech.ac.jp/∼ananta/ca/.

Acknowledgments This work was supported in part by the Global Center of Excellence (GCOE) CompView, Tokyo Institute of Technology.

References

1) Venter, J.C., Adams, M.D., Myers, E.W., et al.: The sequence of the human genome, *Science*, Vol.291, pp.1304–1351 (2001).

2) Waterson, R.H., Lindblad-Toh, K., Birney, E., et al.: Initial sequencing and comparative analysis of the mouse genome, *Nature*, Vol.420, pp.520–562 (2002).

3) Smith, E.E., Buckley, D.G., Wu, Z., et al.: Genetic adaptation by Pseodomonas aeruginosa to the airways of cystic fibrosis patients, *Proc. Natl. Acad. Sci.*, Vol.103, pp.8487–8492 (2006).

4) Eisen, J.A.: Environmental shotgun sequencing: Its potential and challenges for studying the hidden world of microbes, *PLoS Biol.*, Vol.5, e82, DOI:10.1371/journal.pbio.0050082 (2007).

5) Kim, J., Bhinge, A.A., Morgan, X.C. and Iyer, V.R.: Mapping DNA-protein interaction in large genomes by sequence tag analysis of genomic enrichment, *Nat. Methods*, Vol.2, pp.47–53 (2005).

6) Johnson, D.S., Mortazavi, A., Myers, R.M. and Wold, B.: Genome-wide mapping of in vivo protein-DNA interactions, *Science*, Vol.316, pp.1497–1502 (2007).

7) Myers, E.W., Sutton, G.G., Delcher, A.L., et al.: A whole-genome assembly of Drosophila, *Science*, Vol.287, pp.2196–2204 (2000).

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

9) Huang, X., Wang, J., Aluru, S., Yang, S. and Hillier, D.: PCAP: A whole-genome

assembly program, *Genome Res.*, Vol.13, pp.2164–2170.

10) Mullikin, J.C. and Ning, Z.: The Phusion assembler, *Genome Res.*, Vol.13, pp.81–90 (2003).

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

12) Sommer, D.D., et al.: Minimus: a fast, lightweight genome assembler, *BMC Bioinformatics*, Vol.8, 64 (2007).

13) Margulies, M., Egholm, M., Altman, W.E., et al.: Genome sequencing in microfabricated high-density picolitre reactors, *Nature*, Vol.437, pp.476–380 (2005).

14) Bentley, D.R.: Whole-genome re-sequencing, *Curr. Opin. Genet. Dev.*, Vol.16, pp.545–552 (2006).

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

16) Pevzner, P.A., Tang, H. and Waterman, M.S.: An Euler path approach to DNA fragment assembly, *Proc. Natl. Acad. Sci.*, Vol.98, pp.9478–9753 (2001).

17) Dohm, J.C., Lottaz, C., Borodina, T. and Himmelbauer, H.: SHARGS, a fast and highly accurate short read assembly algorithm for *de novo* genomic sequencing, *Genome Res.*, Vol.17, pp.1697–1706 (2007).

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

19) Gallant, J., Maier, D. and Storer, J.A.: On finding minimal length superstrings, *J. Comput. Syst. Sci.*, Vol.20, No.1, pp.50–58 (1980).

20) Turner, J.S.: Approximation algorithm for the shortest common superstring problem, *Information and Computation*, Vol.83, pp.1–20 (1989).

21) Medvedev, P., et al.: Computability of models for sequence assembly. algorithms in bioinformatics, *Lecture Notes in Computer Science*, Vol.4645/2007, pp.289–301 (2007).

22) Pop, M.: Genome assembly reborn: recent computational challenges, *Briefings in Bioinformatics*, Vol.10, No.4, pp.354–366 (2009).

23) Richter, D.C., Ott, F., Auch, A.F., Schmid, R. and Huson, D.H.: MetaSim A sequencing simulator for genomics and metagenomics, *PLoS ONE*, Vol.3, No.10 (2008).

24) Myers, E.W.: Toward simplifying and accurately formulating fragment assembly, *J. Comp. Bio.*, Vol.2, pp.275–290 (1995).

25) Kurtz, S., Phillippy, A., Delcher, A.L., et al.: Versatile and open software for comparing large genomes, *Genome Bio.*, Vol.5, R12 (2004).

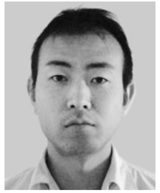
26) Li, S., Zhu, H., Ruan, J., et al.: *De novo* assembly of human genomes with massively parallel short read sequencing, *Genome Res.*, DOI:10.1101/gr/097261.109 (2009).

27) Miller, J.R., Koren, S. and Sutton, G.: Assembly algorithms for next-generation sequencing data, *Genomics*, Vol.95, pp.315–327 (2010).

(Received May 6, 2011)
(Accepted July 12, 2011)
(Released November 4, 2011)

(Communicated by *Jun Sese*)

Wisnu Ananta Kusuma received his M.S. in Software Engineering from Bandung Institute of Technology, Indonesia. Currently, he is a Ph.D. student in the Graduate School of Information Science and Engineering, Tokyo Institute of Technology. He is also a lecturer in Department of Computer Science, Bogor Agriculture University, Indonesia. His current research interest is bioinformatics.



Takashi Ishida is an assistant professor of Graduate School of Information Science and Engineering, Tokyo Institute of Technology. He received his Ph.D. degree from the University of Tokyo, Japan. His current research interest is bioinformatics. He is a member of Information Processing Society of Japan and the Biophysical Society of Japan.

Yutaka Akiyama received Dr. Eng. from Keio University in 1990. He served as the director of Computational Biology Research Center (CBRC), AIST, from 2001 to 2007. He became a professor at Department of Computer Science, Tokyo Institute of Technology in April 2007. His research interest covers parallel processing and computational biology, including genome sequence analysis, protein-protein interaction prediction, mass spectrometry and pharmacokinetics.