Genomics & Informatics Vol. 2(4) 184-190, December 2004

Five Computer Simulation Studies of Whole-Genome Fragment Assembly: The Case of Assembling *Zymomonas mobilis* ZM4 Sequences

Cholhee Jung[1,2], Jin-Young Choi[1], Hyun Seok Park[2,3]* and Jeong-Sun Seo[2,4]

[1]Department of Computer Science and Engineering, Korea University, Seoul 136-701, Korea, [2]Institute of Bioinformatics, Macrogen Corp., Seoul 153-023, Korea, [3]Department of Computer Science and Engineering, Ewha University, Seoul 120-750, Korea, [4]School of Medicine, Seoul National University, Seoul 151-742, Korea

Abstract

An approach for genome analysis based on assembly of fragments of DNA from the whole genome can be applied to obtain the complete nucleotide sequence of the genome of *Zymomonas mobilis*. However, the problem of fragment assembly raise thorny computational issues. Computer simulation studies of sequence assembly usually show some abnormal assemblage of artificial sequences containing repetitive or duplicated regions, and suggest methods to correct those abnormalities. In this paper, we describe five simulation studies which had been performed previous to the actual genome assembly process of *Zymomonas mobilis* ZM4.

Keywords: Shotgun, Fragment assembly, bioinformatics, genome, Zymomonas mobilis

Introduction

Since the random shotgun method[1] was first applied to the whole genome sequencing project of *Haemophilus influenzae* in 1995 (Fleischmann *et al.*, 1995), the shotgun sequencing approach has become the most prevailing method of determining the sequence of a long DNA fragment. The identification of the whole genome sequence of *Zymomonas mobilis* ZM4[2] has also been completed through shotgun method (Seo *et al.*, 2005, in press). The basic random shotgun protocol is as follows:

1) Biologists randomly fracture the sample either using sonication or nebulation.
2) To remove fragments that are too large or small, this pool of fragments is then size-selected.
3) Biologists then insert the size-selected fragments into the vector, to produce bacterial colonies.
4) The sequencing information is processed, reading both end-sequences of each clone.
5) Given the collection of reads obtained from a shotgun protocol, bioinformaticians perform the computational process, called fragment assembly, to infer the source sequence.

We can think of fragment assembly as a jigsaw puzzle[3]. Knowing the approximate length of the target sequence, it is possible to sequence the whole molecule directly. However, we may instead get a piece of the molecule starting at a random position and sequence it in the canonical direction. A fragment corresponds to a substring of one of the strands of the target molecule, but which position in the whole genome would not be known. By using the random shotgun method, we try to reconstruct the target molecule's sequence based on fragment overlap. Fragment assembly is then to deduce the whole sequence of the target DNA molecule. However, this seemingly simple process is not without technical challenges.

In completing the whole genome sequence of

*Corresponding author: E-mail hspark@macrogen.com
Tel +82-2-2113-7007, Fax +82-2-2113-7016
Accepted 20 November 2004

[1] The process of shotgun sequencing begins by physically breaking the DNA into random fragments, which are then read by a sequencing machine. Next, a computer program pieces together the many overlapping reads and reconstructs theorigina sequence.

[2] *Zymomonas mobilis* is an obligatority fermentative Gram-negative bacterium that utilizes sucrose, glucose, and fructose leading to the production of ethanol and CO2. Because *Z.mobili* is an interesting microorganism for its powerful activity of ethanol fermentation, this bacterium has been studied using molecular genetic and biochemical method for last decades (Kang *et al.*, 1998). The full sequence of *Z.mobilis* is registered in NCBI (AE008692).

[3] Usually, sequencing machines are able to handle only 500-1000 base fragments at a time (a read). These fragments must be assembled into a single continuous genomic sequence. There are two broad approaches to this problem: Using an ordered set of markers along a sequence, or using many overlapping sequences (high coverage) to infer ordering directly from the sequences themselves.

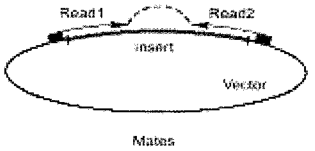


Fig. 1. Mate-pair: Clone inserts are sequenced from both ends, yielding mated sequence reads

Zymomonas mobilis existed many difficult problems. For example, the data usually contains errors, from limitations in sequencing technologies or from human mistakes during laboratory work. Even without experimental errors, fragment assembly have features that complicate the assembly process in a computational sense, most notably, repeats or chimeras. And these problems tend to make spurious contigs[4]. Various algorithms and concepts have been borrowed from computing technology to solve this fragment assembly problem. For example, in step 5, we usually use overlap-layout- consensus[5] strategy (Chen *et al.*, 2000), the most popular approach to fragment assembly. However, this method still makes erroneous assembly because the overlap-layout-consensus strategy has some flaws in handling repetitive or duplicated regions. To overcome these weaknesses, bioinformaticians use mate-pair information[6] (see Fig. 1).

Although hundreds of microbial genomes and several eukaryotic genomes have been sequenced completely, whole-genome sequencing is still one of the major challenges in bioinformatics. Thus, both public and industrial sectors try to improve the assembling process of whole-genome. In this paper, we describe five computer simulation studies of artificial sequences, which had been performed previously to the actual assembly process of *Zymomonas mobilis* ZM4, and which greatly helped us to improve our assembly skills in handling real sequencing data.

Methods

Systems

All our simulations were performed on a linux-box with two Intel PentiumIII-500MHz CPUs and 2-GByte memory.

Table 1. Five source sequences for fragment assembly simulations. Each sequence No. is corresponding to each simulation No.

Seq. No	Length (nt)	Containing repeats	Repeat location
1	100,000	No repeat	-
2	100,000	2k-long 'AAATT'	30,001~32,000
3	100,000	2k-long 'AAATT'	30,001~32,000 60,001~62,000
4	100,000	four identical 1k-long sequences	10,001~11,000 20,001~21,000 30,001~31,000 70,001~71,000
5	100,000	three identical 5k-long sequences	10,001~15,000 20,001~25,000 60,001~65,000

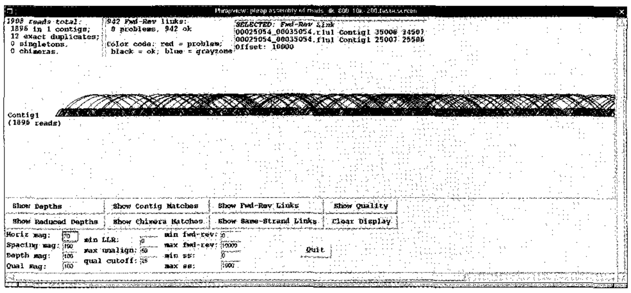
For simulations, we initially used **phrap** (Green, 1999) as a fragment assembly tool. However, we had to develop several analysis softwares additionally, which are written in Python, Perl and C++.

Studies on some repeat patterns

We started simulations with randomly generated 100kb-long artificial sequence. Then we added 4 different repeat patterns to the original sequence to analyze various results of assembly, corresponding to each repeat pattern. Characteristics of the five 100kb-long sequences are described in the Table 1. Repeats could be sorted in two groups. One is tandem repeats with short repeat unit and the other is duplication of identical sequences.

In all simulations, both 2k fragments and 10k fragments were generated. The number of 2k fragments is 800 and the number of 10k fragments is 200, so that total number of fragments is 1,000. Each fragment is composed of two reads: front and rear. So the total number of reads is 2,000. We assumed that the read-length is 500nt. Therefore, sequencing coverage in each simulation is 10x fold. After every simulation, we checked the accuracy of assembly result with the **phrapview** of the phrap package and **bl2seq** of NCBI.

Simulation No. 1: The first simulation is on an artificial



```
1908 reads total:          942 Fwd-Rev links:             SELECTED: Fwd-Rev Link
 1896 in 1 contigs;         0 problems, 942 ok           00025054_00035054.r1ul Contig1 35006 34507
 12 exact duplicates;                                    00025054_00035054.f1ul Contig1 25007 25506
 0 singletons.             Color code: red = problem;    Offset: 10000
 0 chimeras.                black = ok; blue = grayzone

Contig1
(1896 reads)

Horiz mag:   70                        min fwd-rev: 0
Spacing mag: 100  min LLR:      0      max fwd-rev: 10000
Depth mag:   100  max unalign:  50     min ss:      0
Qual mag:    100  qual cutoff:  25     max ss:      1000
```

Fig. 2. Location of each read and its mate-pair in the contig. Edegs (or arches) represent mate pairs.

[4] Contig: A contiguous region of DNA sequence assembled from overlapping reads of random cloned fragments

[5] Overlap Phase: Every read is compared to every other read. Layout Phase: Relative position of the reads is determined, and spanning forest of overlaps is used to produce a layout. Consensus Phase: Multi-alignment of reads in regions where coverage is greater than 3.

[6] Mate Pair: DNA sequence reads from opposite ends of a single cloned piece of DNA (see Fig. 1). It could come from small insert clone (2 kbp), medium insert clone (10 kbp), or BAC clone. It is used for computational gap closure

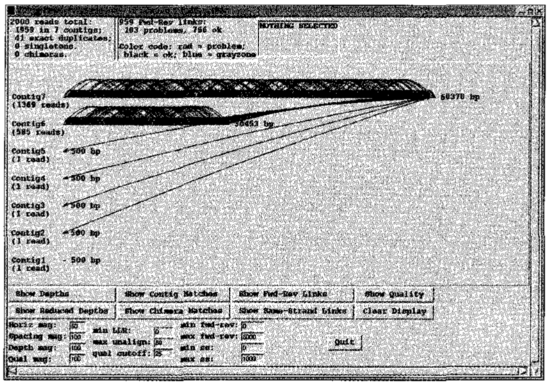
Table 2. Mate-pair information between Contig6 and Contig7.

```
Contig 6 opp sense RIGHT LINK:   complement Contig 7
* Seq100k_s2-00022080_00032080.f1u1
  Seq100k_s2-00022080_00032080.r1u1 8906 22080 67845
* Seq100k_s2-00022557_00032557.f1u1
  Seq100k_s2-00022557_00032557.r1u1 8906 22557 67368
  ......
```

sequence of length 100kb. As described above, the total number of 2,000 reads were generated and assembled. Assembling the sequences resulted in only one contig of length 99,953nt. For the sake of simplicity, we assume that short tandem repeats are not included in the alignments. In Fig. 2, the phrapview[7] shows that all the fragments constructing the contig are arranged appropriately. We also checked the accuracy by comparing the assembled contig with the original source sequence using bl2seq[8] (Fig. 9-a).

Simulation No. 2: This simulation is processed with sequence number 2. As shown in Table 1, this sequence contains one repetitive region. In this simulation, fragments were assembled into 7 contigs (Fig. 3). Among them, only two contigs could be assembled only with the right mate pairs, however, the rest of them consisted of 'taaat', 'aaatt', 'attaa', 'aatta', or 'ttaaa', respectively. Now, we have to make sure whether the two longer contigs could be ordered by mate-pair information. For this, we referenced '.phrap.out' file, one of the phrap result files. This file contains mate-pair information of inter or intra contigs. As in table 2, mate-pair information indicates that the reverse-complemented Contig7 comes after the Contig6, and this ordering is confirmed by the pair-wise alignment results (See Fig. 9-b and Fig. 9-c). Table 2 also shows that the size of the gap between two longer contigs is about 1.1k, and this gap-size information is useful for additional gap-closing process.

Simulation No. 3: In this simulation, the original sequence was separated into 3 regions by the two repetitive regions. So, the phrap made 6 short contigs and 4 long contigs out of collection of reads (See Fig. 4). Six short contigs are almost same with each other, that is, they are filled with repetitive patterns. Repetitve pattern is also shown in both the rear regions of Contig9 and one end of the Contig8 and Contig10 (See Fig. 5). Although the rear regions of contigs are not clearly assembled, 10k-mate information gives us appropriate ordering information.

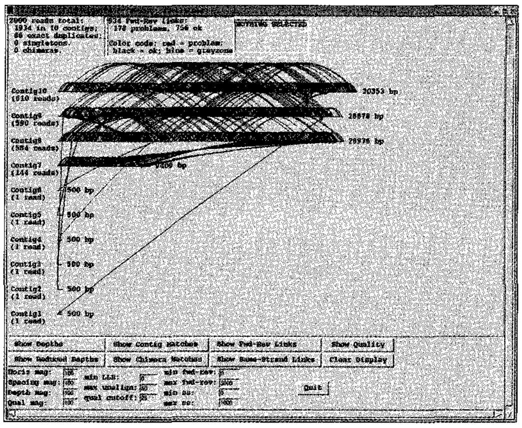


```
2000 reads total:
 1959 in 7 contigs;
 41 exact duplicates;
 0 singletons.
 0 chimeras.
```

```
959 Fwd-Rev links:
 183 problems, 766 ok

Color code: red = problem;
 black = ok; blue = grayzone
```

NOTHING SELECTED

Fig. 3. Assembly result of simulation No 2. Five among seven contigs just consist of repetitive patterns.

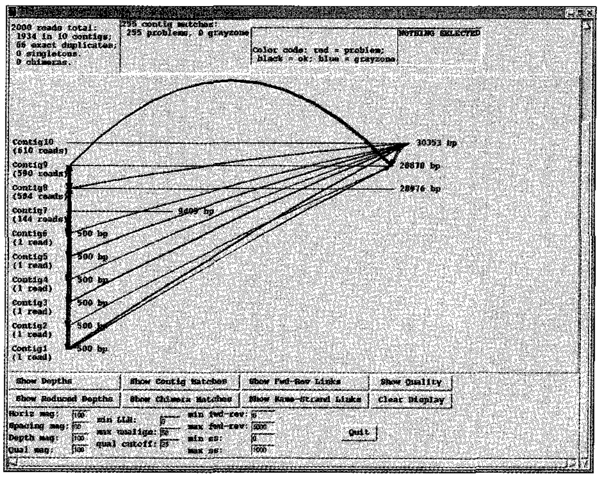


```
2000 reads total:
 1934 in 10 contigs;
 66 exact duplicates;
 0 singletons.
 0 chimeras.
```

```
934 Fwd-Rev links:
 178 problems, 756 ok

Color code: red = problem;
 black = ok; blue = grayzone
```

NOTHING SELECTED

Fig. 4. Assembly result of simulation No 3: the original sequence was separated into 3 regions by the two repetitive regions. There are 6 short contigs and 4 long contigs out of collection of reads.



```
2000 reads total:
 1934 in 10 contigs;
 66 exact duplicates;
 0 singletons.
 0 chimeras.
```

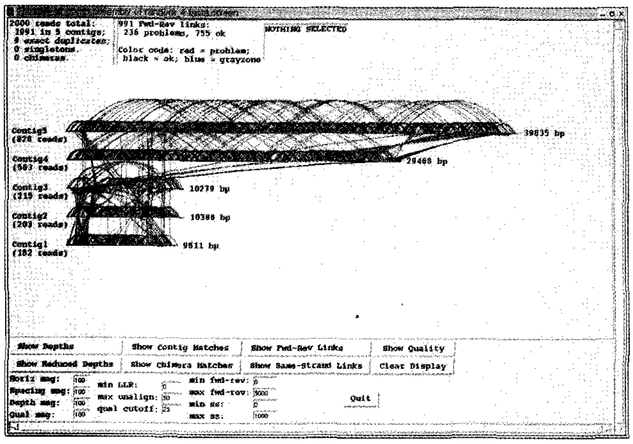
```
255 contig matches:
 255 problems, 0 grayzone

Color code: red = problem;
 black = ok; blue = grayzone
```

NOTHING SELECTED

Fig. 5. Similar regions among 10 contigs. All contigs have repetitive region at least. Six short contigs are almost same with each other, that is, they are filled with repetitive patterns. Repetitve pattern is also shown in both the rear regions of Contig9 and one end of the Contig8 and Contig10.

[7] Phrapview is a graphical tool that provides a "global" view of the phrap assembly.

[8] Bl2seq performs a comparison between two sequences using either the blastn or blastp algorithm. Both sequences must be either nucleotides or proteins.

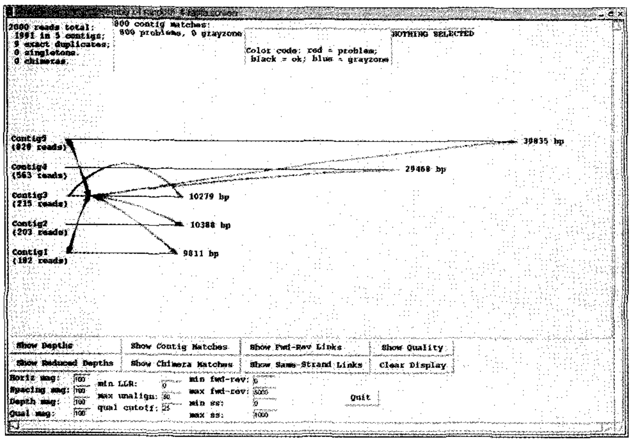


```
2000 reads total:        991 Fwd-Rev links:
 1991 in 5 contigs;       236 problems, 755 ok
 9 exact duplicates;
 0 singletons.           Color code: red = problem;
 0 chimeras.              black = ok; blue = grayzone
```

NOTHING SELECTED

```
Horiz mag:   100   min LLR:     0    min fwd-rev: 0
Spacing mag: 100   max unalign: 50   max fwd-rev: 5000
Depth mag:   100   qual cutoff: 25   min ss:      0
Qual mag:    100                     max ss:      1000
```

Fig. 6. Assembly result of simulation No 4. In the blue circle in the left side of the figure, 2k-mate information is not feasible, but *10k-mate information covers the weakness of 2k-mate information.*

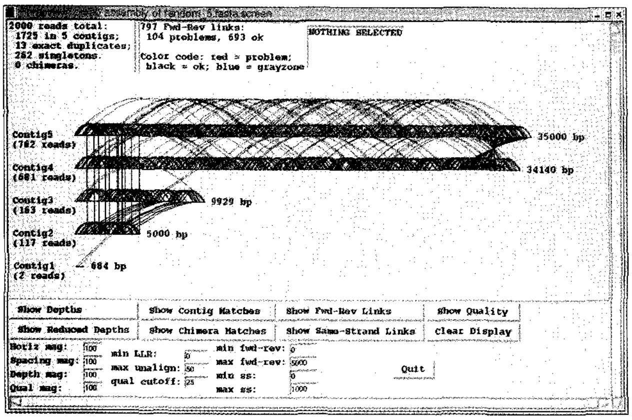


```
2000 reads total:        800 contig matches:
 1991 in 5 contigs;       800 problems, 0 grayzone
 9 exact duplicates;
 0 singletons.           Color code: red = problem;
 0 chimeras.              black = ok; blue = grayzone
```

NOTHING SELECTED

```
Horiz mag:   100   min LLR:     0    min fwd-rev: 0
Spacing mag: 100   max unalign: 50   max fwd-rev: 5000
Depth mag:   100   qual cutoff: 25   min ss:      0
Qual mag:    100                     max ss:      1000
```

Fig. 7. Similarity information among all contigs.



```
2000 reads total:        797 Fwd-Rev links:
 1725 in 5 contigs;       104 problems, 693 ok
 13 exact duplicates;
 262 singletons.         Color code: red = problem;
 0 chimeras.              black = ok; blue = grayzone
```

NOTHING SELECTED

```
Horiz mag:   100   min LLR:     0    min fwd-rev: 0
Spacing mag: 100   max unalign: 50   max fwd-rev: 5000
Depth mag:   100   qual cutoff: 25   min ss:      0
Qual mag:    100                     max ss:      1000
```

Fig. 8. Assembly result of simulation No 5: All the five contigs, except Contig1, seemed to be well assembled because of their feasible mate-pair information.

So, we could re-arrange 4 longer contigs in correct order (Fig. 10-a), and this ordering was confirmed with the pair-wise alignment result (Fig. 9-d).

Simulation No. 4: Contrary to the simulation number 2 and number 3, this simulation contains four identical sequences which is randomly generated (Table 1). After running phrap, we obtained five contigs. All the contigs have feasible mate-pair information (Fig. 6). By using mate-pair information, we could finally arrange the five contigs in appropriate order (Fig. 10-b). In fact, 2k mate-pair information in blue circle in Fig. 6 is not evenly spread. However, 10k mate information covers the region with evenly overlapped mate-pairs information.

Simulation No. 5: In the previous simulation, it seems not to be problematic to assemble sequence which has relatively short identical sequences (in this case, 1kb). However, many biological sequences contain a few identical regions longer than 1kb. So, the fifth simulation used more realistic sequence close to the original data (Table 1). In this simulation, the phrap assembled the reads into 5 contigs. All the five contigs, except Contig1, seemed to be well assembled because of their feasible mate-pair information (See Fig. 8). Table 3 shows that several 10k-mate pair information, Contig4-Contig5, Contig5-Contig2 and Contig2-Contig3 were adjacent to each other. However, the estimated gap-sizes are all 5k, although each mate-pair information linking Contig4-Contig5, Contig5-Contig2 and Contig2-Contig3 is 10k-mate. This means that each gap should contain one of three identical 5k sequences so that the gap-size could be 10k. Therefore, these contigs could be ordered as shown in Fig. 10-c.

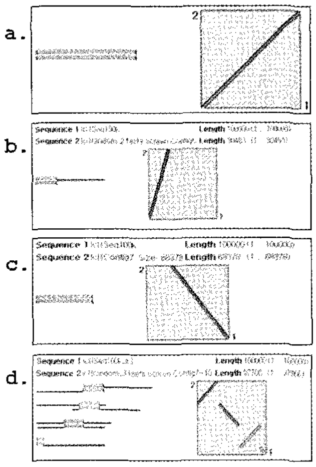


Fig. 9. pair-wise alignment results. In the left corner of each figures, upper bar is the original sequence and the lower one is anassembled sequence. 9-a is from simulation No 1, and it shows that there is no inconsistency between the original sequence and the assembled sequence. 9-b and 9-c is, respectively, the pair-wise alignment results of Contig6 and Contig7 of the simulation No 2. And the last one, 9-d is from simulation No 3.

Table 3. Mate-pair information linking Contig4-Contig5, Contig5-Contig2 and Contig2-Contig3.

Contig 5 opp sense RIGHT LINK: complement Contig 4				
Seq100k_s4-00055733_00065733.f1u1	Seq100k_s4-00055733_00065733.r1u1	5000	30733	33407
Seq100k_s4-00056211_00066211.f1u1	Seq100k_s4-00056211_00066211.r1u1	5000	31211	32929
Contig 4 opp sense RIGHT LINK: complement Contig 5				
Seq100k_s4-00059865_00069865.r1u1	Seq100k_s4-00059865_00069865.f1u1	5000	29275	34865
Seq100k_s4-00059620_00069620.r1u1	Seq100k_s4-00059620_00069620.f1u1	5000	29520	34620
Contig 3 opp sense RIGHT LINK: Contig 2				
Seq100k_s4-00005696_00015696.f1u1	C Seq100k_s4-00005696_00015696.r1u1	4999	5625	695
Seq100k_s4-00006013_00016013.f1u1	C Seq100k_s4-00006013_00016013.r1u1	4999	5942	1012
Contig 2 opp sense LEFT LINK: Contig 3				
C Seq100k_s4-00005696_00015696.r1u1	Seq100k_s4-00005696_00015696.f1u1	4999	695	5625
C Seq100k_s4-00006013_00016013.r1u1	Seq100k_s4-00006013_00016013.f1u1	4999	1012	5942
Contig 2 opp sense RIGHT LINK: Contig 5				
Seq100k_s4-00015949_00025949.f1u1	C Seq100k_s4-00015949_00025949.r1u1	4999	949	948
Seq100k_s4-00016489_00026489.f1u1	C Seq100k_s4-00016489_00026489.r1u1	4999	1489	1488

Table 4. (a) fragment sizes from the predicted physical map from (Kang *et al.,* 1998) and (b) the actual sizes from fragment assembly in (Seo *et al.*, 2005)

*Pme*I		*Pac*I	
a.(kb)	b.(nt)	a.(kb)	b.(nt)
625	656,065	525	526,958
331	330,536	229	218,361
195	195,609	191	186,419
191	189,027	132	132,006
159	158,750	132	131,331
138	140,008	112	114,317
85	82,779	110	105,931
69	65,914	100	95,865
65	62,094	100	95,726
60	58,735	100	93,386
37	36,341	96	91,456
36	35,249	83	82,518
35	33,993	76	73,952
9	8,457	47	46,828
3	2,859	32	29,865
		18	16,221
		9	8,492
		7	6,780

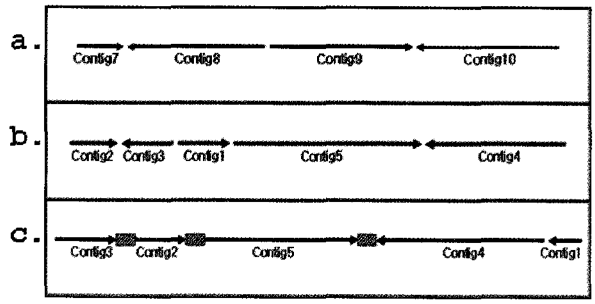


Fig. 10. Contig ordering based on the pate-pair information. 10-a, 10-b and 10-c is the ordering result of the simulation No 3, No 4 and No 5, respectively. In 10-c, three gray boxes indicate the three identical 5k sequences.

Assembly process of Zymomonas mobilis

Still an open question in bioinformatics would be how to classify all the repeat patterns in a genomics sequence. One of the difficulties in repeat classification is that many repeats represent mosaics of sub-repeats (Bailey *et al.*, 2002). Different combinations of sub-repeats form different repeat copies (Bao *et al.*, 2002).

Although this problem looks easy to solve for the toy examples as in the five simulations, inconsistencies in local alignments make it extremely difficult for real genomics sequences. Through the five simulations, we studied how we could apply mate-pair information to ordering contigs in real data. Now, we will describe the process of assembling *Zymomonas mobilis* by applying the methods we used in the simulations.

Initial assembly result

The best known programs for repeat annotation are RepeatMasker module in Phrad package, which use precompiled repeat libraries to find copies of known repeat families represented in RepBase. However, the repeat libraries have to be manually compiled for any new genome like *Zymomonas mobilis*, because they are genome-specific. Thus, the only possible approach to repeat analysis of a newly sequenced genome would be to simply list all pairs of repeated regions.

At first, tens of thousands of read data were picked. After filtering out wrongly sized mate-pair and chimera[9], approximately 42,000 reads were selected and assembled into 436 contigs (Fig. 11). The edges represent mate pairs. Contigs are assembled after filtering out all the possible repeats, using both the existing RepeatMasker and domestically developed software tools, written in Perl language.

Assembling into longer contigs

The initial stage of single linkage clustering approach

[9] Chimera usually means an organism or recombinant DNA molecules created by joining DNA fragments from two or more different organisms.

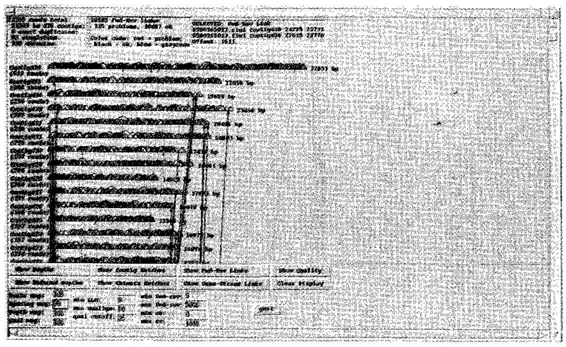


Fig. 11. Initial assembly result of *Zymomonas mobilis* 42,000 reads. Mate-pair information is relatively well overlapped in alomost all the contigs.

starts from finding pairwise similarities and use clustering to group similar sequences together. However, clustering based on sequence similarities presents a serious problem because local sequence alignments do not typically correspond to the biological boundaries of a repeat (Bao, *et al.*, 2002)

In real sequencing data, the sequencing quality is unreliable in the front region of every sequencing data, and this low-correctness appears in the rear region of sequencing data too. So, some read data could be assembled into several different contigs, although they could be merged into one in the end. Therefore, if one contig's end region is very similar to the other, the two contigs could be merged into one longer contig. However, this concept is not used in our simulations; we assumed that every read has no low-quality (thus, unreliable) region. In the initial version of assembly, we merged 436 contigs into initial version of eight longer contigs. However, the newly assembled eight contigs still had to be re-checked for their feasibility.

Mate-pair information was used for consistency. If a region consists of contiguously overlapping mate-pairs, that region is supposed to be well assembled. This idea is derived from the simulation No 4. From the previous simulation results, we further split 8 contigs into 49 stable contigs (or unitigs[10]).

The iterative splitting procedure converges to a graph composed solely of rightful mate pair edges only. The best way to split suspected edges is to remove one of its low-multiplicity edges because high-multiplicity edges typically connect the most conserved positions in the repeat (Pevzner, *et al.*, 2004).

Ordering and scaffolding

For the following process, we used 384 FOSMID clones

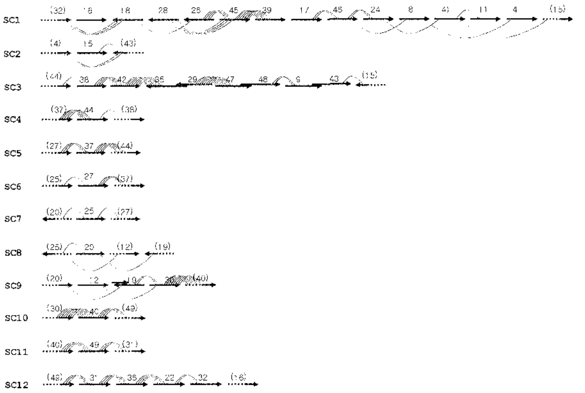


Fig. 12. Ordering contigs into 12 scaffolds. Each blue arrow indicates one of 49 refined contigs, and red arcs indicates FOSMID mate-pair information.

as long-ranged mate-pair information. As we ordered contigs by using mate-pair information in our simulations, we ordered 49 contigs into 12 scaffolds[11] successfully (Fig. 12). For additional ordering information, we used a physical map of *Zymomonas mobilis* genome (Kang *et al.*, 1998). In Kang *et al.*, A physical map of the *Z. mobilis* ZM4 genome was constructed by aligning *Pme*I fragments and *Pac*I fragments with each other by reciprocal Southern blot hybridization. Seven genomic *Not*I fragments were also used (Fig. 16-a). As in Fig. 16-b, we could arrange 12 scaffolds in appropriate order by using the physical map. However, the physical map in (Kang *et al.*, 1998). was slightly different from our final result, but the differences between Fig. 13-a and 13-b could be left out of account (Table 4).

Annotation of *Zymomonas mobilis*

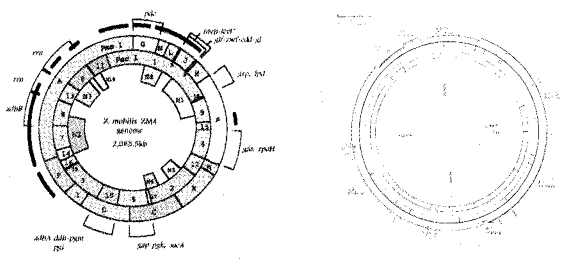
After the ordering process had been completed, we closed all gaps through PCR or sometimes through additional shotgun. Finally, we analyzed the completed genome sequence to annotate the characteristics of the genome sequence. The final length of the completed genome sequence was 2,056,416bp. Detailed analysis result of annotation is presented in (Seo *et al.*, 2005, in press).

Result and Discussion

Assembly software aims to make correct contigs and then re-arrange the contigs in appropriate order. But, real biological sequences contain some patterns of repeat that make assembly process extremely complicated. In this paper, we simulated fragment assembly processes

[10] Uniquely Assemble-able Contig

[11] A set of ordered BAC or FOSMID clones linking contigs togethe



(a) (b)

Fig. 13. (a) A Physical map of *Z.mobilis* genome shown in Kang *et al.* (1998) and (b) computer-generated physical map from fragment assembly. The physical map indicates three restriction enzymes, rrn opreons and several important genes. Three restriction enzymes in this map are *PmeI*, *PacI* and *NotI*.

of re-constructing artificial sequences, intentionally adding various repeats to compare the various assembly results corresponding to different repeat patterns. The simulation results show that the mate-pair information could play key roles, even though the original sequence has repeats. Especially in simulation number 4, mate-pair information was the key to assembly confirmation. Implicitly, this infers that mate-pair information could be used for refining contigs in the case that contigs have mal-assembled regions. In the confirming process of the initial version of *Zymomonas mobilis* assembled sequence, the initially assembled eight contigs had to be split further into 49 refined contigs. They had to be re-assembled in the later process, in a constantly repetitive fashion.

Some recently developed assembly softwares use mate-pair information so that it makes contig sequence more accurate (Batzoglou *et al.*, 2002). We chose Phrap and Arachne for a rigorous benchmarking study. But, spurious contigs were still inevitable - even though the existing assembly softwares adopted mate-pair information, it still makes an assembly problem if the source sequence is long and contains various repeats. Biological editors of genome sequence have to observe the mate-pair information of assembly results carefully, and modify the results in a repetitive fashion to constantly improve the final result of assembly.

We believe that the techniques shown in our simulations and in assembly process of the *Z.mobilis* could be helpful not only for performing the following genome sequencing projects but also for improving the performance in developing assembly softwares.

Acknowledgements

We are indebted to Chunnan Chen of Applied Biosystems, for his invaluable advice on assembling *Zymomonas mobilis*. This work is partially supported under the grant number of IMT2000 ab-02 and ab-05 by the Korean Ministry of Information and Communication.

References

Bailey, J.A., Yavor, A.M., Viggiano, L., Misceo, D., Horvath, J.E., Archidiacono, N., Schwartz, S., Rocchi, M., and Eichler, E.E. (2002). Human-specific duplication and mosaic transcripts: The recent paralogous structure of chromosome 22. *Am. J. Hum. Genet.* 70, 83-100.

Bao, Z. and Eddy, S. (2002). Automated de novo Identification of repeat sequence families in sequenced genomics. *Genome Res.* 8, 1269-1276.

Batzoglou, S., Jaffe, D.B., Stanley, K., Butler, J., Gnerre, S., Mauceli, E., Berger, B., Mesirov, J.P., and Lander, E.S. (2002). Arachne: A whole-genome shotgun assembler. *Genome Res.* 12, 177-189.

Chen, T. and Skiena, S. (2000). A case study in genome-level fragment assembly. *Bioinformatics* 16, 494-500.

Fleischmann, R.D., Adams, M.D., White, O., Clayton, R.A., Kirkness, E.F., Kerlavage, A.R., et al. (1995). Whole-Genome Random Sequencing and Assembly of *H. influenzae*. *Science* 269, 496-512.

Green, P., Documentation for phrap, *http://bozeman.mbt.washington.edu/phrap.docs/phrap.html.*

Kang, H.L. and Kang, H.S. (1998). A physical map of the genome of ethanol fermentative bacterium Zymomonas mobilis ZM4 and localization of genes on the map. *Gene* 206, 223-228.

Pevzner, P.A., Tang, H., Tesler, G. (2004). De Novo Repeat Classification and Fragment Assembly. *Genome Research* 14, 1786-1796.

Seo, J.S., Chong, H., Park, H.S., Yoon, K.O., Jung, C., Kim, J.J., *et al.* (2005). The genome sequence of the ethanologenic bacterium *Zymomonas mobilis* ZM4. *Nature Biotechnology*, in press.