Bioinformatics and Biosystems 2006, Vol. 1, No. 1, pp. 61-66

MLP: Mate-based Sequence Layout with PHRAP

Jinwook Kim[1], Kangho Roh[1], Kunsoo Park[1],
Hyunseok Park[2], Jeongsun Seo[2]

[1]School of Computer Science and Engineering, Seoul National University, Seoul, Korea
[2]Macrogen, Seian Building, 116, Shinmun-Ro 1Ka, Chongro-Ku, Seoul, Korea

Abstract

We propose a new fragment assembly program MLP (mate-based layout with PHRAP). MLP consists of PHRAP, repeat masking, and a new layout algorithm that uses the mate pair information. Our experimental results show that by using MLP instead of PHRAP, we can significantly reduce the difference between the assembled sequence and the original genome sequence.

Introduction

The aim of fragment assembly programs is to find the genome sequence from sequence reads. Sequence reads are the fragments generated by shotgun sequencing. Various assemblers have been developed for this purpose such as PHRAP (Green, 1994), TIGR (Sutton et al., 1995), AMASS (Kim et al., 1999), CAP3 (Huang et al., 1999), STROLL (Chen et al., 2000), the Celera assembler (Myers et al., 2000), ARACHNE (Batzoglou et al., 2002), and RePS (Wang et al., 2002).

Most of assemblers adopt the overlap-layout-consensus strategy. The overlap procedure finds the pairwise alignments of the reads. Next, the layout procedure determines the order and the position of the reads among them. Finally, the consensus procedure finds the base in each position.

In the layout procedure, the mate pair information is very useful. The mate pair information comes from the way we read bases in an insert. An insert is a small fragment which is a randomly partitioned piece of a genome. Although inserts with lengths 2 kbp, 10 kbp, 40 kbp, etc. are possible, we consider only ones with 2 kbp in this paper. From the both ends of each insert, about 500 bp are read by sequencers - these two reads are the mates of each other.

The procedure of PHRAP is follows. It gets the information of reads from the data file, then finds the exact matches for every pair of reads. Based on the exact matches, PHRAP constructs pairwise alignments of reads. These alignments are inexact matches and have LLR scores (Green, 1994). PHRAP sorts all aligned pairs in descending order of the LLR scores. Then, PHRAP constructs a contig for each read and merges the contigs by the sorted order of the aligned pairs. Finally, the consensus sequences are made.

PHRAP has been widely used for fragment assembly, but it has two drawbacks. First it does not resolve well the repeat problem, and second it does not use the mate pair information in its assembly though mate pairs are specified in its input.

One of the latest assemblers, RePS (Wang et al., 2002), uses PHRAP as it is, but adds two processes: a pre-PHRAP process that identifies exactly repeated 20mers and masks them out, and a post-PHRAP process that constructs scaffolds from contigs using the mate pair information.

We have combined PHRAP, repeat masking, and a new layout algorithm that uses the mate pair information into a fragment assembly program, MLP (Mate-based Layout with PHRAP). We dived into the source code of PHRAP and modified the layout part (data structures and layout procedure) so that the mate pair information can be used during the layout. The overlap and consensus procedures of PHRAP remain the same. In addition, MLP uses repeat masking as RePS does. See Figure 1. Our experiments show that MLP produces significantly better sequencing results than PHRAP.

The rest of this paper is organized as follows. Section 2 describes the methods of MLP and section 3 provides experimental results of MLP and PHRAP. Finally, section 4 gives a conclusion.

Corresponding Author: Kunsoo Park (Tel: 02-880-8381, Fax: 02-885-3141, Email: kpark@theory.sun.ac.kr)
This work was supported by the BK 21 Project and the IMT 2000 Project AB02.

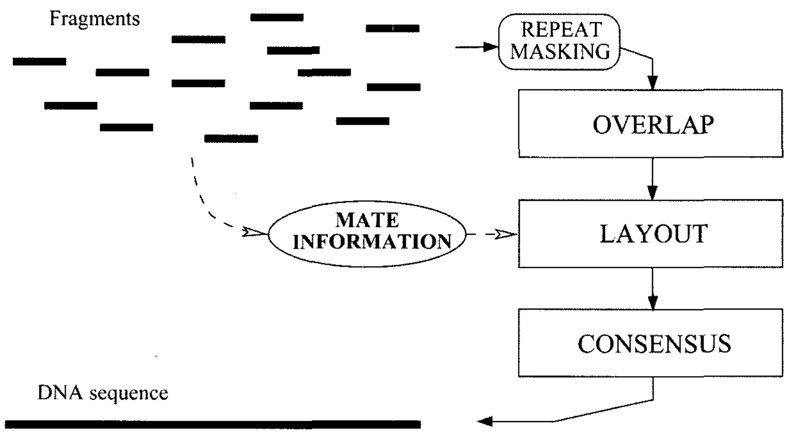

Figure 1 : Overview of MLP. We modified the data structures and the layout procedure of PHRAP so that the mate pair information can be used during the layout. The overlap and consensus procedures of PHRAP remain the same. In addition, MLP uses repeat masking as RePS does.

Methods

Before describing the methods of MLP, we give some notations. Let a, b, $\cdots$ and a', b', $\cdots$ denote the reads, where reads a, a' and reads b, b' are mate pairs.

At first, one contig is constructed for one mate pair. SeeFigure 2. Between two reads of a mate pair, there is a gap with length 1kbp. Thus there is also a gap in the contig. Of course, if a read does not have a mate read, a contig is constructed for only one read.

During MLP we always maintain the condition that both reads of a mate pair are in one contig. In other words, If read a is in contig A and has a mate read a', then read a' is also in contig A. This condition is met initially.

We now explain our strategies to merge contigs. To merge contigs, there must be some similarity between them. We consider three cases that are used in the layout procedure. In each case, read a is in contig A and read b is in contig B:

Case 1: Two mate pairs, a, a' and b, b', are aligned, i.e., a and b are aligned and also a' and b' are aligned. See Figure 3.

Case 2: Reads a and b are aligned, but their mate reads are not aligned. However, read b' and read c that is already in contig A and is lain in a similar location with a' are aligned. See Figure 4 (i).

Case 3: Reads a and b are aligned, but their mate reads are not aligned. However, reads d and e that are already in contigs A and B and are lain in similar locations with a' and b', respectively, are aligned. See Figure 4 (ii).

We give more explanations about three cases. In Case 1, since two mate pairs are aligned, the two contigs are similar in at least two regions. Thus these contigs are a good candidate to merge. In Case 2, read c is lain in a similar location with a and read c is already in contig A. It means that reads c and a' have some similarity. Thus reads a' and b' also have some similarity. In Case 3, reads d and a' have some similarity and reads e and b' also have some similarity.

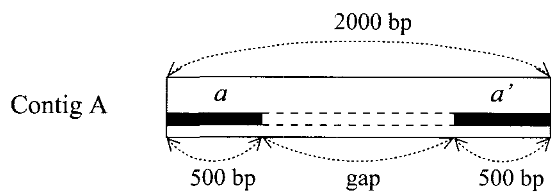

Figure 2. Initial contig. Contig A is constructed for reads a and a' which are a mate pair.

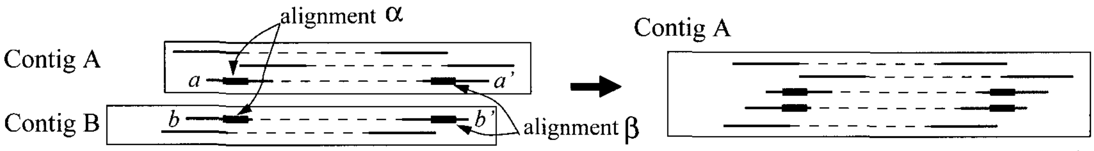

Figure 3. Case 1 - two mate pairs, a , a' and b , b', are aligned, i.e., a and b are aligned and also a' and b' are aligned.

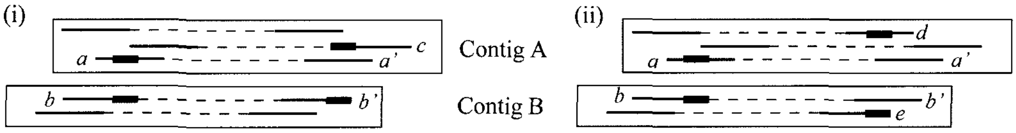

Figure 4. (i) Case 2 - reads a and b are aligned, but their mate reads are not aligned. However, read b' and read c that is already in contig A and is lain in a similar location with a' are aligned. The symmetric case is also possible (i.e., the alignment of read a' and some read that is already in contig B and is lain in a similar location with b'). (ii) Case 3 - reads a and b are aligned, but their mate reads are not aligned. However, reads d and e that are already in contigs A and B and are lain in similar locations with a' and b', respectively, are aligned.

Because of the alignment of reads d and e, reads a' and b' have some similarity, too.

In each case, we allow variation in gap lengths of the contigs. Because of inaccuracy of experiments, the lengths of inserts may not be exactly 2 kbp. In other words, when the length of each read in a mate pair is 500 bp, the gap length between them may not be exactly 1 kbp. Thus we assume that the lengths of inserts are normally distributed around the mean value of 2 kbp with a standard deviation of 200 bp.

We need to test the candidate contigs for consistency before merging. See Figure 5. We find out all alignments between two contigs. After that, the LLR score of the alignments and the gap length between the adjacent alignments are checked. In addition, we check the gap length of each insert in candidate contigs. Because the lengths of inserts may not be the same, the distance between alignments α and β of contig A and the distance between α and β of contig B in Case 1 of Figure 3 may be different. This situation can happen in Case 2 and Case 3 as well. In this situation we should change the gap length in one contig and check the insert length. See Figure 6. We use the maximum variation 1500 bp in the insert size. The range 2 kbp±1500 bp covers over 99% of all the inserts under the 10% standard deviation.

We made two versions of the layout algorithm of MLP. Figure 7 shows Version 1 of the layout algorithm. Suppose that the average insert size is 2 kbp and the average read size is 500 bp. Initially, we construct a contig with 1 kbp gap for each mate pair. Then all aligned pairs are sorted in descending order of the LLR scores (Green, 1994). We merge contigs through the following three steps. Each step runs for all aligned pairs.

In the first step, we test whether the current aligned pair satisfies Case 1. If it satisfies, then the contigs are merged together after **test_merge**(). In the second step, we test Case 1, Case 2, and Case 3. If any one of them is true, then the contigs are merged together after **test_merge**(). The third step is the original layout procedure of PHRAP. In this step we do not consider the alignment of the mate reads, but consider only one aligned pair.

Version 2 of the layout algorithm is almost same as Version 1. The difference is that in step 3 (b) in Figure 7, we check only Case 1 and Case 2. Let k_a and k_b be the number

---

1. Check the minimum LLR score of alignments.
2. Check the maximum gap length between alignment segments.
3. Check the gap length of each insert. The maximum variation in the insert size is 1500 bp.

Figure 5: **test_merge**() of MLP

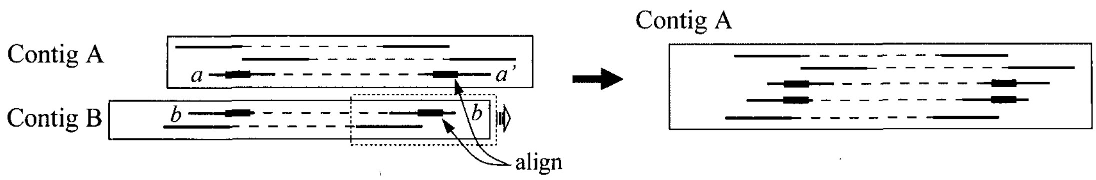

Figure 6. Variable gap length. The gap length in one contig is changed.

---

1. Construct contigs with 1 kbp gap for each mate pair.
2. Sort all aligned pairs in descending order of the LLR scores.
3. Merge contigs through the following three steps. Each step runs for all aligned pairs.
   (a) If the current aligned pair satisfies Case 1, then the contigs are merged together after **test_merge()**. In this step, the maximum variation in the gap length of a contig is 100 bp.
   (b) If the current aligned pair satisfies Case 1, Case 2, or Case 3, then the contigs are merged together after **test_merge()**. The maximum variation is 200 bp.
   (c) Run the original layout of PHRAP.

Figure 7. Layout algorithm of MLP, Version 1

of reads which are lain in a similar location with read a' and read b', respectively. To test Case 2, we need to find the alignments between a' and each of the k_b reads and between b' and each of the k_a reads. To test Case 3, we need to find the alignments between each of the k_a reads and each of the k_b reads. Hence, Case 2 checks $k_a + k_b$ alignments and Case 3 checks $k_a \times k_b$ alignments. Therefore, Version 2 runs faster than Version 1, while its sequencing result is a little worse than that of Version 1.

Results

We had experiments with MLP and PHRAP on three simulated data sets from *Homo sapiens* chromosome 11 clone RP11-701I24 complete sequence (GenBank Acc. No. AC090707 (187 Kbp)), *Brucella suis* 1330 chromosome II

Table 1. Experimental results. Set 1 is *Homo sapiens* chromosome 11 with length 187 kbp, Set 2 is *Brucella suis* 1330 chromosome II with length 1224 kbp and Set 3 is *Homo sapiens* chromosome 17 with length 146 kbp. The coverage of the sets are 10-fold. For each set, we list the number of contigs and the Diff value. Diff is one of the CROSS_MATCH results and is the difference between the assembled sequence and the original genome sequence.

		MLP		PHRAP
		Version 1	Version 2	
Set 1	Number of contigs	36	46	38
	Diff	135238	138682	139791
Set 2	Number of contigs	9	9	9
	Diff	12416	12416	12612
Set 3	Number of contigs	69	199	77
	Diff	1501449	1625635	2635925

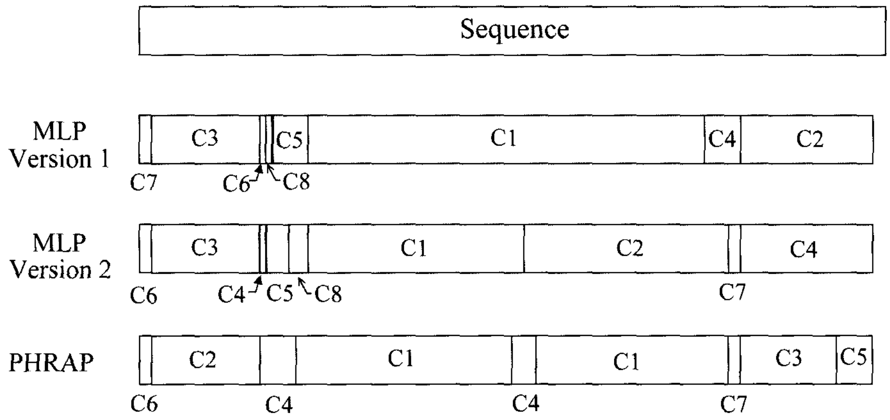

Figure 8. The CROSS_MATCH results of Set 1. (*Homo sapiens* chromosome 11 with length 187 kbp) The first box named sequence is the target genome sequence. The boxes named start with C are the contigs which are the results of each assembler. In case PHRAP, there are two C1s and C4s. It means that they are parts of same contig C1 or C4 but they must be separated.

(GenBank Acc. No. NC_004311 (1224 Kbp)) and *Homo sapiens* chromosome 17 clone RP11-764D10 complete sequence (GenBank Acc. No. AC103703 (146 Kbp)).

The average insert length is 2 kbp and the coverage of the simulated data sets is 10-fold. We made simulated data sets as follows. First, we cut out fragments with length 2 kbp. Then, we cut 500 bp with 10% error from each side of a fragment. The 10% error means that at each position of 500 bp we change the base with probability 0.1, i.e., delete it, duplicate it, or replace it by another base.

The experimental results are shown in Table 1. It shows the number of contigs and the Diff value of CROSS_MATCH for each set (Green, 1994). The Diff value is the difference between the assembled sequence and the original genome sequence, and it is the sum of insertions, deletions, and mismatches in all alignments between them. Our experimental results show that by using MLP Version 1 instead of PHRAP, we can significantly reduce the number of contigs and the Diff value. In case of Version 2, though the number of contigs may be larger than that of PHRAP, the Diff values are reduced.

Figure 8 shows the alignments of the result contigs of MLP and PHRAP with the original target sequence Set 1. It is done by CROSS_MATCH. The box named "sequence" is the target genome sequence and the boxes named "C*i*" are the contigs which are the results of each assembler. We omit the contigs with small lengths. C1 is the longest contig of each assembler, C2 is the next one, and so on. The longest contig C1 produced by PHRAP is aligned with the target sequence at two separate locations. It means that C1 of PHRAP is misassembled. C4 of PHRAP is misassembled, too. In case of Version 2, C4 is misassembled. But in case of Version 1, there is no misassembled contig.

Discussion

MLP is a fragment assembly program designed to exploit the mate pair information in sequencing. MLP has been tested on simulated data from human chromosomes and brucella suis chromosomes and it gives better results than PHRAP. It significantly reduces the difference between the assembled sequence and the original genome sequence.

PHRAP does not have a procedure to resolve the repeat problem. While MLP also has no such procedure, but MLP produces better results than PHRAP by using the mate pair information. We are now developing an advanced fragment assembly program which includes some methods to solve the repeat problem.

References

[1] Batzoglou, A., Jaffe, D. B., Stanley, K., Butler, J., Gnerre, S., Mauceli, E., Berger., B., et al. (2002) ARACHNE: A Whole-Genome Shotgun Assembler. *Genome Research*. 12: 177-189.

[2] Chen, T. and Skiena, S. S. (2000) A case study in genome-level fragment assembly. *Bioinformatics*. 16:

494-500.
[3] Green, P. (1994) PHRAP documentation. http://www.phrap.org.
[4] Huang, X. and Madan, A. (1999) CAP3: A DNA Sequence Assembly Program. *Genome Research.* 9: 868-877.
[5] Kim, S. and Segre, A. M. (1999) AMASS: A Structured Pattern Matching Approach to Shotgun Sequence Assembly. *Journal of Computational Biology.* 6: 163-186.
[6] Myers, E. W., Sutton, G. G., Delcher, A. L., Dew, I. M., Fasulo, D. P., Flanigan, M. J., Kravitz, S. A., et al. (2000) A Whole-Genome Assembly of Drosophila. *Science.* 287: 2196-2204.
[7] Sutton, G. G., White, O., Adams, M. D. and Kerlavage, A. R. (1995) TIGR Assembler: A New Tool for Assembling Large Shotgun Sequencing Projects. *Genome Science Technology.* 1: 9-19.
[8] Wang, J., Wong, G. K., Ni, P., Han, Y., Huang, X., et al. (2002) RePS: A Sequence Assembler That Masks Exact Repeats Identified from the Shotgun Data. *Genome Research.* 12: 824-831.