Hindawi Publishing Corporation
Journal of Biomedicine and Biotechnology
Volume 2008, Article ID 675741, 7 pages
doi:10.1155/2008/675741

Research Article

A Scaffold Analysis Tool Using Mate-Pair Information in Genome Sequencing

Pan-Gyu Kim,[1] Hwan-Gue Cho,[2] and Kiejung Park[1]

[1] *SmallSoft Co., Ltd., Jang-Dong 59-5, Yusung-Gu, Daejeon 305-343, South Korea*
[2] *Department of Computer Science and Engineering, Pusan National University, Busan 609-735, South Korea*

Correspondence should be addressed to Kiejung Park, kjpark@smallsoft.co.kr

Received 1 September 2007; Accepted 24 December 2007

Recommended by Daniel Howard

We have developed a Windows-based program, *ConPath*, as a scaffold analyzer. *ConPath* constructs scaffolds by ordering and orienting separate sequence contigs by exploiting the mate-pair information between contig-pairs. Our algorithm builds directed graphs from link information and traverses them to find the longest acyclic graphs. Using end read pairs of fixed-sized mate-pair libraries, *ConPath* determines relative orientations of all contigs, estimates the gap size of each adjacent contig pair, and reports wrong assembly information by validating orientations and gap sizes. We have utilized ConPath in more than 10 microbial genome projects, including *Mannheimia succiniciproducens* and *Vibro vulnificus*, where we verified contig assembly and identified several erroneous contigs using the four types of error defined in *ConPath*. Also, *ConPath* supports some convenient features and viewers that permit investigation of each contig in detail; these include contig viewer, scaffold viewer, edge information list, mate-pair list, and the printing of complex scaffold structures.

1. INTRODUCTION

In 2001, the Human Genome Project (HGP) Consortium and Celera Genomics reported the first drafts of sequences of the human genome [1, 2]. The HGP Consortium used the hierarchical sequencing or "clone-by-clone" approach, whereas Celera Genomics used the whole genome shotgun (WGS) approach, which had been successfully used in 1995 to sequence the *H. influenzae* genome [3].

In the hierarchical sequencing approach, a tiling of large DNA sequences, such as bacterial artificial chromosome (BAC) or yeast artificial chromosome (YAC), are constructed for a genome, and each of the sequences is determined. The HGP Consortium used BAC as the large sequence, followed by shotgun sequencing of each BAC.

In sequencing the genome, owing to physical limitations of shotgun sequencing methods, the genome must be broken down into smaller portions, shotgun reads sized in the range of 600 bps (base-pairs) to 800 bps, and as the sequence data for each of these shotgun reads is produced, it must be connecting them with those adjacent and overlapping reads that have been previously sequenced, that is, to achieve an assembly of these smaller sequences into larger contiguous regions or "contigs."

In most cases, the sequences of shotgun reads are obtained by sequencing both ends of a DNA fragment whose approximate size is known. Such pair information, referred to as mate-pair information, constrains the placement of the reads within an assembly. In an ideal assembly, all read pairs are placed in such a manner as to satisfy the orientation and distance constraints imposed by the pairing. Mate-pair information can be used to determine the quality of an assembly, because most types of misassemblies lead to violations of these constraints.

In contrast to hierarchical sequencing, WGS breaks a whole genome into small pieces randomly, without shearing into large DNA pieces of intermediate size. WGS is faster and cheaper than hierarchical sequencing because of the simplicity of the processing steps. The success of WGS [4, 5] has increased its usage and the size of the genome to be sequenced has increased.

Although contig assembly programs are well established, less is known about scaffold analysis. While some of its features have been implemented to sequence specific genomes

[6–8], the features needed for general scaffold analysis and visualization have not been provided. *Consed* [9], a graphical tool for contig assembly, provides good visualization and helps to finish sequencing by connecting with *Autofinish* [10]; however, it does not have many features related to scaffold analysis.

It has been suggested that the contig scaffolding problem can be solved by *greedy-path merging algorithm* [8]. Moreover, *GigAssembler* can orient the contigs based on mRNA, paired plasmid ends, EST, and BAC end pairs [7].

This paper introduces a novel scaffold analysis tool, *ConPath*, which calculates the longest scaffolds. Due to the abundance of repeats in genomic DNA sequences, a purely overlap-based approach for WGS assembly is not feasible, but the use of mate-pair information is crucial. The *ConPath* program uses end read pairs of fixed-sized DNA libraries as mate-pairs to calculate orientations, orders, and gap sizes. It reads a *Phrap* [11] output file (∗.out) and an *ACE* format file, which contain contig structures and mate-pair information.

2. MATERIALS AND METHODS

2.1. Mate-pair information

The most important characteristic of *ConPath* is its ability to exploit the mate-pair information of large DNA fragments such as fosmids or cosmids, which are about 40 kbps(kilo base-pairs) in size, or BACs, which are about 100–300 kbps in size, rather than plasmids, which are about 2–10 kbps in size. Figure 1 shows an example of mate-pair end reads. A mate-pair is composed of two end reads that always face each other. Each end read, b or g, has an orientation relative to the contig containing it. If the direction of an end read is the same as the direction of the contig, the former has direction U, otherwise, it has direction C. In Figure 1, b has direction U because the C_l contig and b read are in the same direction, whereas g has direction C because the C_2 contig and g read are in opposite directions. The size of the mate-pair helps to estimate the gap size between contigs C_1 and C_2. When one contig contains one end of a mate-pair and a second contig contains the other end of the mate-pair, the two contigs are said to be linked by the mate-pair. A scaffold is a series of contigs that can be linked by mate-pairs. The connection relationship of all the contigs can be represented as a graph in which each contig is represented as a vertex. An edge is created between two contig vertices when they are linked by at least one mate-pair, and the number of linking mate-pairs between two contigs is defined as the edge weight.

2.2. Construction of scaffolds

To construct scaffolds using mate-pair information, a scaffold graph can be defined as follows.

Given a set of contigs $C = \{c_1, c_2, c_3, \ldots, c_n\}$, a mate-pair set $M = \{m_1, m_2, m_3, \ldots, m_l\}$, and a set of reads $R = \{r_1, r_2, r_3, \ldots, r_s\}$, let G denote the scaffold graph using C and M:

$$G = (C, E). \tag{1}$$

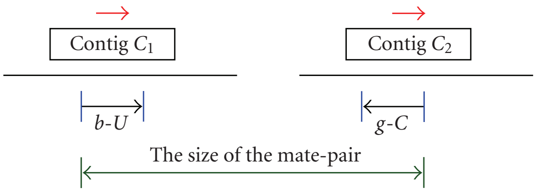


Figure 1: An example of mate-pair information. Mate-pair reads are indicated as read "b" and "g" and the relative directions to encompassing contigs are denoted as "U (same direction)" and "C (complementary direction)."

When a mate-pair $m_k = (r_i, r_j)$ exists, in which contig c_s contains r_i and contig c_t contains r_j, there is an edge between contigs c_s and c_t. Edge set E is expressed as

$$E = \{e_{c_s c_t} \text{ iff } m_k = (r_i, r_j) \text{ exists for } r_i \in c_s,\\ r_j \in c_t, c_s \in C, \text{and } c_t \in C\}. \tag{2}$$

In constructing a scaffold graph, the linking level (l), the threshold value for the edge weights, was used as a filtering value in constructing and showing scaffolds on output. When an edge has a weight value smaller than the linking level (l), the edge is discarded from the graph.

Considering the errors that occur in base calling and contig assembly, the optimal construction of a scaffold graph is an NP-complete problem [8]. To practically solve this problem, *ConPath* uses a simple greedy algorithm. Whenever a new edge is added to the graph, graph G is additive modified for that edge. This provides a feasible heuristic solution for a scaffold construction in linear time. Algorithm 1 shows the algorithm of *ConPath* to construct scaffolds.

2.3. Determination of the orders and orientations of contigs

It is worthwhile noting that *ConPath* determines the relative orientations of all contigs using the orientations of the end reads.

Figure 2 shows the determination of the order and orientations of three contigs using two mate-pairs. In Figure 2(a), b_1 and g_1 reads determine the relative orientation of contigs C_1 and C_2, and, in the same way, b_2 and g_2 reads determine the relative orientations of contigs C_2 and C_3 (see Figure 2(b)). The relative orientations of contigs C_1, C_2, and C_3 are determined by rotating the scaffold in Figure 2(b), as shown in Figure 2(c).

2.4. Estimation of the gap size between contigs

Assuming all mate-pairs have a fixed size, the size of the gap between two adjacent contigs is determined by the sizes of the two contigs and the positions of the end reads of contigs.

Suppose that contig C_1 contains b read and contig C_2 contains g read. Let Gap (C_1, C_2) be the gap size between C_1 and C_2. Let $P_s(b)$ and $P_e(b)$ be the start and end positions of b read in C_1, respectively, and let $P_s(g)$ and $P_e(g)$ be the start

Table 1: Mate-pair information in real test datasets. The proportion of mate-pair reads for *V. vulnificus* is about double that for *M. succiniciproducens*.

Genome	Genome length	Fold	Number of reads	Number of mate-pairs	Proportion of mate-pair reads relative to number of reads
M. succiniciproducens	2.3 Mbp	13.2	about 25,000*	275	2.2%
V. vulnificus	5.1 Mbp	11.7	76,971	1,781	4.5%

* The numbers of reads for 4 versions of *M. succiniciproducens* show slight variation.

Table 2: Real test datasets. Four datasets for the *M. succiniciproducens* genome and one for the *V. vulnificus* genome were tested with *ConPath*. MP: mate-pair, MPIC: mate-pair in the same contigs.

Data name	Number of contigs	Number of MPs	Number of MPICs	Average size of MP(fosmid)s
MH1	98	238	72	37,673 bp
MH2	86	240	115	38,102 bp
MH3	85	240	120	38,157 bp
MH4	112	240	108	37,917 bp
VV	334	1,220	454	33,024 bp

```
MakeScaffold(mate-pair set)
{
  initial scaffold graph $\mathbf{G} = (\mathbf{V}, \mathbf{E}; \mathbf{V} = \{\text{all contigs}\}, \mathbf{E} = \{\})$
  assign each mate-pair to corresponding an edge → edge set $\mathbf{E}$
  remove self-collision mate-pairs
  while (edge set $\mathbf{E}$ is not empty)
      find an edge with maximum weight from $\mathbf{E} \rightarrow \mathbf{e}(\mathbf{k}, 1)$
      if ($\mathbf{e}(\mathbf{k}, 1)$ does not conflict with $\mathbf{G}$)
         add $\mathbf{e}(\mathbf{k}, 1)$ to graph $\mathbf{G}$
      delete $\mathbf{e}(\mathbf{k}, 1)$ from edge set $\mathbf{E}$
  end while
}
```

Algorithm 1: The algorithm for scaffold construction. *ConPath* uses a simple greedy algorithm to obtain a feasible heuristic solution for an NP-complete problem.

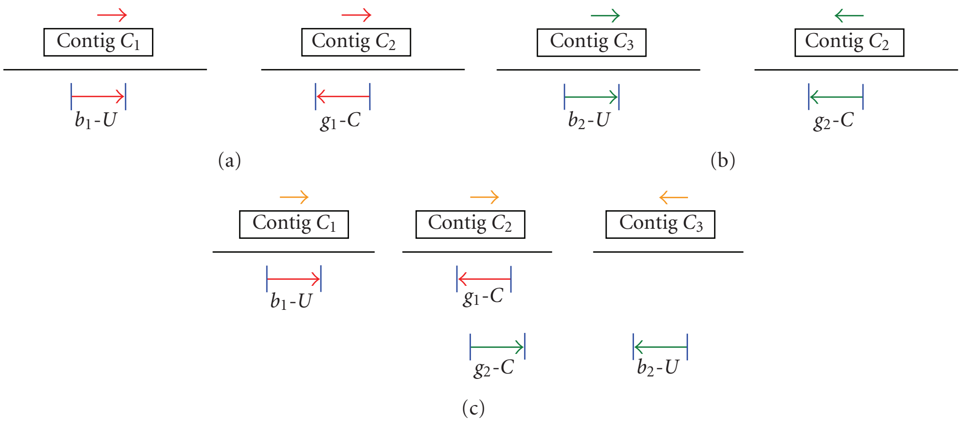


Figure 2: Determining the relative orientations of contigs using mate-pair information. (a): b_1 and g_1 reads determine the relative orientation of contigs C_1 and C_2; (b): b_2 and g_2 reads determine the relative orientations of contigs C_2 and C_3; and (c): the relative orientations of contigs C_1, C_2, and C_3 are determined by rotating the scaffold in Figure 2(b).

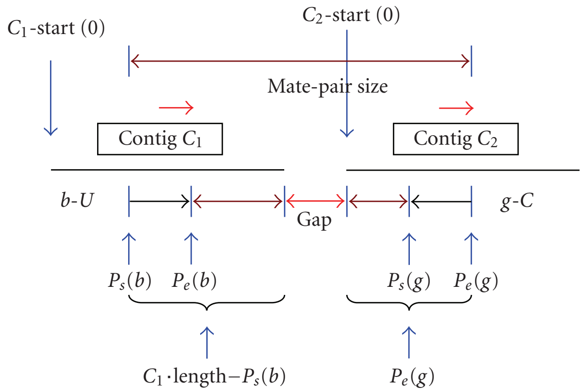
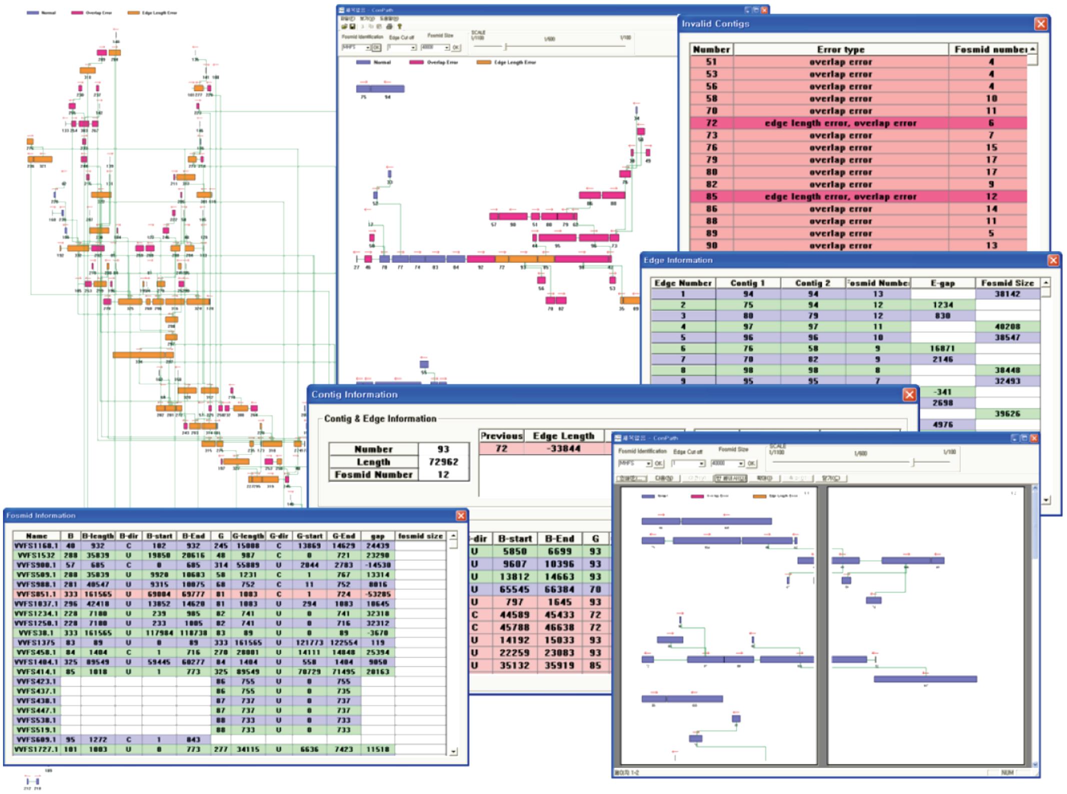


Figure 3: Estimation of the gap size between contigs when b has direction U and g has direction C. The gap size between C_1 and C_2 can be calculated as mate_pair size $- \{(C_1 \cdot \text{length} - P_s(b)) + P_e(g)\}$.



Invalid Contigs

Number	Error type	Fosmid number
51	overlap error	4
53	overlap error	4
56	overlap error	4
58	overlap error	10
70	overlap error	11
72	edge length error, overlap error	6
73	overlap error	7
76	overlap error	15
79	overlap error	17
80	overlap error	17
82	overlap error	9
85	edge length error, overlap error	12
86	overlap error	14
88	overlap error	11
89	overlap error	5
90	overlap error	13

Edge Information

Edge Number	Contig 1	Contig 2	osmid Numbe	E-gap	Fosmid Size
1	94	94	13		38142
2	75	94	12	1234	
3	80	79	12	830	
4	97	97	11		40208
5	96	96	10		38547
6	76	58	9	16871	
7	70	82	9	2146	
8	98	98	8		38448
9	95	95	7		32493
				-341	
				2698	
					39626
				4976	

Contig Information

Contig & Edge Information

Number	93
Length	72962
Fosmid Number	12

Previous	Edge Length
72	-33844

-dir	B-start	B-End	G
U	5850	6699	93
U	9607	10396	93
U	13812	14663	93
U	65545	66384	70
U	797	1645	93
C	44589	45433	72
C	45788	46638	72
U	14192	15033	93
U	22259	23083	93
U	35132	35919	85

Fosmid Information

Name	B	B-length	B-dir	B-start	B-End	G	G-length	G-dir	G-start	G-End	gap	fosmid size
VVFS1168.1	40	932	C	102	932	245	15008	C	13869	14629	24439	
VVFS1532	288	35839	U	19850	20616	48	987	C	0	721	23290	
VVFS900.1	57	685	C	0	685	314	55889	U	2044	2783	-14530	
VVFS509.1	288	35839	U	9920	10683	58	1231	C	1	767	13314	
VVFS988.1	281	40547	U	9315	10075	68	752	C	11	752	8016	
VVFS851.1	333	161565	U	69004	69777	81	1083	C	1	724	-53285	
VVFS1037.1	296	42418	U	13852	14620	81	1083	U	294	1083	10645	
VVFS1234.1	228	7180	U	239	985	82	741	U	0	741	32318	
VVFS1250.1	228	7180	U	233	1005	82	741	U	0	716	32312	
VVFS38.1	333	161565	U	117984	118738	83	89	U	0	89	-3670	
VVFS1375	83	89	U	0	89	333	161565	U	121773	122554	119	
VVFS458.1	84	1404	C	1	716	270	28001	U	14111	14848	25394	
VVFS1404.1	325	89549	U	59445	60277	84	1404	U	558	1404	9050	
VVFS414.1	85	1018	U	1	773	325	89549	U	70729	71495	20163	
VVFS423.1						86	755	U	0	755		
VVFS437.1						86	755	U	0	735		
VVFS438.1						87	737	U	0	737		
VVFS447.1						87	737	U	0	737		
VVFS538.1						88	733	U	0	733		
VVFS519.1						88	733	U	0	733		
VVFS609.1	95	1272	C	1	843							
VVFS1727.1	101	1003	U	0	773	277	34115	U	6636	7423	11518	

Figure 4: A set of snapshots of *ConPath*. *ConPath* provides a set of useful information, "mate-pair information", "edge information", "contig path", and "invalid contigs" by checking for the 4 types of error.

and end positions of g read in C_2, respectively. Considering all the possible directions of a mate-pair of two end reads, *ConPath* estimates the gap size as

b-U and g-U: Gap (C_1, C_2) = mate_pair size $- \{(C_1 \cdot \text{length} - P_s(b)) + (C_2 \cdot \text{length} - P_s(g))\}$

b-U and g-C: Gap (C_1, C_2) = mate_pair size $- \{(C_1 \cdot \text{length} - P_s(b)) + P_e(g))\}$

b-C and g-U: Gap (C_1, C_2) = mate_pair size $- \{P_e(b) + (C_2 \cdot \text{length} - P_s(g))\}$

b-C and g-C: Gap (C_1, C_2) = mate_pair size $- \{P_e(b) + P_e(g)\}$

Figure 3 shows the procedure for estimating the gap size between contigs when b and g have U and C directions,

TABLE 3: Number of reported errors in scaffold construction for 5 dataset.

Data name	Errors*	l			
		1	2	3	4
MH1	Self Collision	0	0	0	0
	Gap size	3	3	3	0
	Overlap	22	2	0	2
MH2	Self collision	2	2	2	2
	Gap size	2	2	2	2
	Overlap	20	2	2	0
MH3	Self collision	0	0	0	0
	Gap size	5	0	0	0
	Overlap	18	0	0	0
MH4	Self collision	0	0	0	0
	Gap size	0	0	0	0
	Overlap	0	0	0	0
VV	Self collision	16	16	10	7
	Gap size	65	7	3	2
	Overlap	85	24	0	4

* Mate-pair size errors were excluded because these errors do not depend on l.

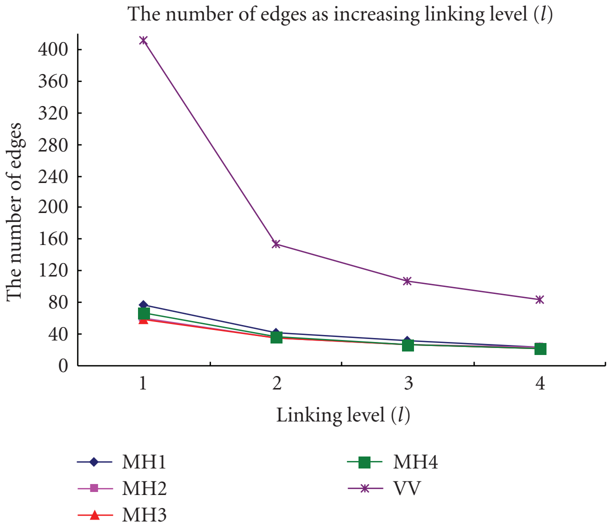


FIGURE 5: Distribution of the number of edges according to linking level (l). *ConPath* constructed the best scaffolds at linking level 2 while minimizing edge loss.

respectively. The orientations of contigs C_1 and C_2 are set in the same direction. The length of part of the mate-pair library in contig $C_1(C_1 \cdot \text{length} - P_s(b))$ and the length of part of the mate-pair library in contig $C_2(P_e(g))$ are calculated. Finally, the gap size is calculated as

$$\text{mate\_pair size} - \{(C_1 \cdot \text{length} - P_s(b)) + P_e(g)\}. \tag{3}$$

2.5. Detection of erroneous contigs

One important feature of *ConPath* is the verification of a contig assembly by identifying erroneous contigs. We have defined 4 types of contig assembly errors to check the quality of a contig assembly.

Self-collision error

When the number of mate-pairs connecting two adjacent contigs is more than 2, and there is an inconsistency in determining the orientation of contigs with mate-pairs, the error is defined as a self-collision error, the most serious error type. If this error occurs, the contigs should be inspected manually one by one.

Mate-pair size error

When a mate-pair of an end read is contained in a contig, the real size of this mate-pair can be calculated. If the difference between the calculated and predefined sizes is larger than a threshold value, the error is defined as a mate-pair size error. This type of error is very critical to the contig assembly process.

Gap-size error

If the gap size between two contigs is a negative value, it indicates that the two contigs should be merged in the contig assembly process; this is defined as a gap size error.

Overlap error

After calculating the distances of all adjacent contigs, any two nonadjacent contigs can be overlapped due to the accumulation of errors in gap size estimations. This type of error is defined as an overlap error, which happens rarely and is not so critical.

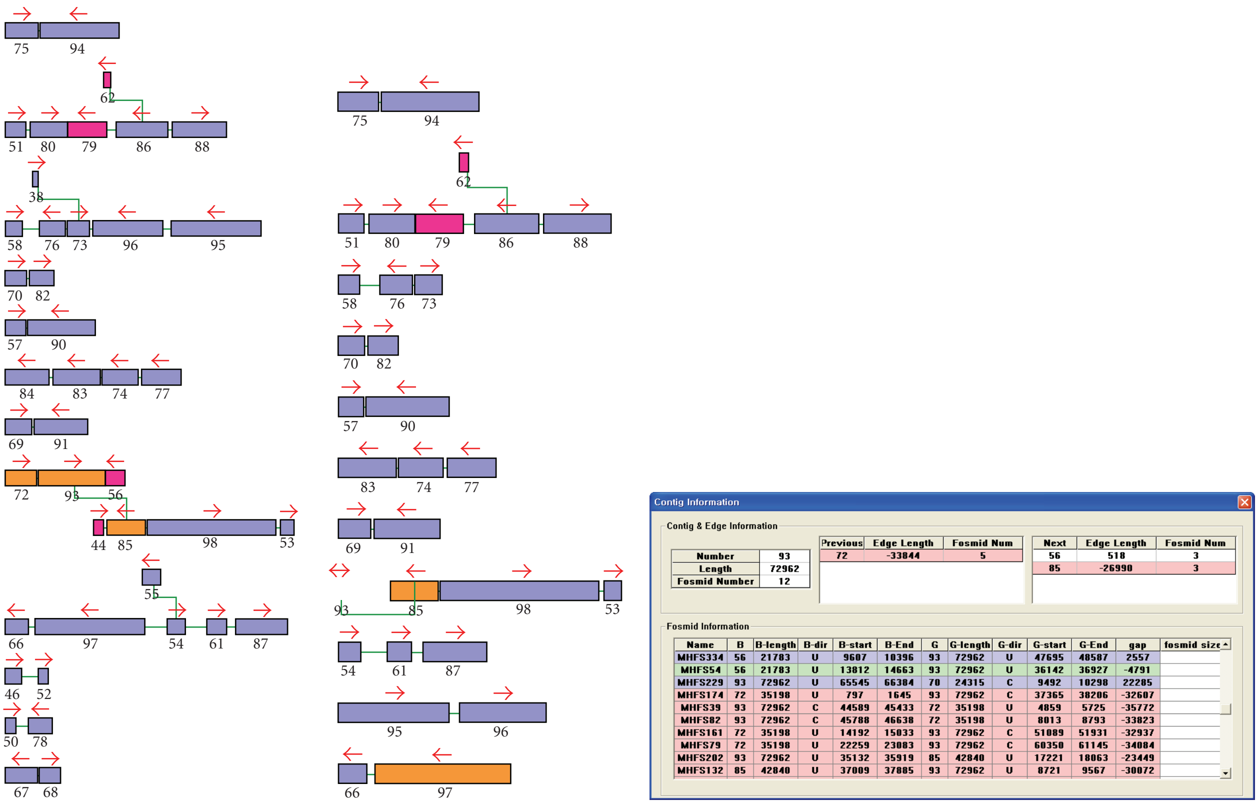
Identifying error types is useful in verifying and correcting the final result of a contig assembly. If a contig has more than two types of errors, it is highly probable that a misassembled contig is present. *ConPath* assigns different colors to contigs by the number of error types, with nonerroneous contigs colored blue. When one contig has more than one error, *ConPath* assigns this contig a reddish color, with the intensity proportional to the number of error types. Therefore, we can check the quality of the final result of a contig assembly by simply inspecting the color information in the scaffold visualization window of *ConPath*.

2.6. Implementation

ConPath was implemented on a Windows XP system using Visual C++. It provides a user-friendly interface and shows visual and color-informative outputs, which can help analyze scaffolds both intuitively and informatively. *ConPath* provides dialogue windows for “mate-pair information”, “edge information”, “contig path”, and “invalid contigs” by automatically checking for the 4 types of errors. Scaffolds are displayed graphically in proportion to the real sizes of vertices and edges after aligning vertices and edges to avoid graphical collision, and the detailed information for each vertex and edge is shown on a pop-up window. *ConPath*

Table 4: Comparison of *ConPath* with other scaffold tools.

Comparison item	Tools			
	ConPath	*Consed*	*Autofinish*	*Bambus*
Accuracy of scaffold	Medium	Medium	Medium	Strong
Construction time	Strong	Strong	Strong	Strong
Visualization	Strong	Medium	Weak	Weak
Error detection	Strong	Medium	Medium	Medium
Additional information	Strong	Strong	Medium	Medium



Contig Information

Contig & Edge Information

Number	93
Length	72962
Fosmid Number	12

Previous	Edge Length	Fosmid Num
72	-33844	5

Next	Edge Length	Fosmid Num
56	518	3
85	-26990	3

Fosmid Information

Name	B	B-length	B-dir	B-start	B-End	G	G-length	G-dir	G-start	G-End	gap	fosmid size
MHFS334	56	21783	U	9607	10396	93	72962	U	47695	48587	2557	
MHFS54	56	21783	U	13812	14663	93	72962	U	36142	36927	-4791	
MHFS229	93	72962	U	65545	66384	70	24315	C	9492	10298	22285	
MHFS174	72	35198	U	797	1645	93	72962	C	37365	38206	-32607	
MHFS39	93	72962	C	44589	45433	72	35198	U	4859	5725	-35772	
MHFS82	93	72962	C	45788	46638	72	35198	U	8013	8793	-33823	
MHFS161	72	35198	U	14192	15033	93	72962	C	51089	51931	-32937	
MHFS79	72	35198	U	22259	23083	93	72962	C	60350	61145	-34084	
MHFS202	93	72962	U	35132	35919	85	42840	U	17221	18063	-23449	
MHFS132	85	42840	U	37009	37885	93	72962	U	8721	9567	-30072	

(a) (b) (c)

Figure 6: An example of the detection of mis-assembled contigs. (a): Scaffolds for MH1 at linking level 2; (b): scaffolds for MH1 at linking level 3; (c): information on contig 93.

can produce a large picture for all scaffolds by assembling separately printed module pictures. Figure 4 shows various viewers and dialogues of *ConPath*.

3. EXPERIMENTS AND DISCISSION

We tested *ConPath* using both artificial and real data. Artificial data were generated in two different versions: *R* (randomly) and *U* (uniformly). The *R* version consisted of contigs of random sizes, whereas the *U* version consisted of contigs of uniform size. In these artificial data experiments, *ConPath* showed very successful scaffold constructions using mate-pair information. From experiments with artificial data, *ConPath* made a reasonable scaffold construction in linear time.

ConPath worked very successfully and efficiently on real data sets, in sequencing the *Mannheimia succiniciproducens* and *Vibro vulnificus* genomes. *ConPath* verified the results of contig assembly by detecting misassembled contigs. Table 1 shows the mate-pair information in these real datasets. Four datasets were tested in sequencing the *M. succiniciproducens* genome, whereas one dataset was tested in sequencing the *V. vulnificus* genome, to verify the results of contig assembly. Table 2 shows these results. MH1, MH2, MH3, and MH4 are the contig assembly results of the *M. succiniciproducens* genome and VV is the contig assembly result for the *V. vulnificus* genome. For the *M. succiniciproducens* genome, going from MH1 to MH4 increased the reliability of the contig assembly results.

We examined the edge number according to linking level (see Figure 5). *ConPath* was most successful at linking level 2 by minimizing the loss of edges.

Table 3 shows the detected errors in scaffold construction for the 5 datasets. Among the *M. succiniciproducens* datasets, MH1 had the most errors, whereas MH4 had no erroneous contigs. These results show that identifying the 4 types of errors for contigs is effective in verifying the result of contig assembly.

Figure 6 shows the constructed scaffolds at linking levels 2 and 3 for the MH1 dataset. Contig 93 is suspected of being erroneous because it has several erroneous contigs on both sides. *ConPath* showed that contig 93 was misassembled. The contig information dialogue box for contig 93 is shown in Figure 6(c).

Table 4 shows a comparison of features of several scaffold analysis tools, including *ConPath*, *Consed* [9], *Autofinish* [10], and *Bambus* [12]. Compared with these other tools, *ConPath* has very good features for 5 criteria. Most importantly, *ConPath* helps users to intuitively verify the contig assembly by providing many visualization features and additional information to detect erroneous contigs.

4. CONCLUSION

A scaffold analyzer is a very important tool in genome sequencing, in that it can verify the results of contig assembly and to identify misassembled contigs. We have developed *ConPath*, a scaffold analyzer that exploits mate-pair information to construct scaffolds by ordering and orienting separate sequence contigs. *ConPath* provides various useful viewers and dialogue boxes for intuitive understanding. Using end read pairs of a fixed-sized mate-pair library, *ConPath* can determine the relative orientations of all contigs successfully, and estimate the gap size of each adjacent contig pair. We defined 4 types of errors to detect misassembly. *ConPath* was used successfully in sequencing several microbial genomes, including the *M. succiniciproducens* genome [13]. *ConPath* is, therefore, a useful scaffold analyzer to verify contig assembly by detecting erroneous contigs.

ConPath will doubtless improve as its algorithm becomes more correct and efficient, as well as through the development of additional features, such as primer design for the finishing step and a sequence read viewer.

REFERENCES

[1] E. S. Lander, L. M. Linton, B. Birren, et al., "Initial sequencing and analysis of the human genome," *Nature*, vol. 409, pp. 860–921, 2001.

[2] J. C. Venter, M D. Adams, E. W. Myers, et al., "The sequence of the human genome," *Science*, vol. 291, no. 5507, pp. 1304–1351, 2001.

[3] R. D. Fleischmann, M. D. Adams, O. White, et al., "Whole-genome random sequencing and assembly of *Haemophilus influenzae Rd*," *Science*, vol. 269, no. 5223, pp. 496–512, 1995.

[4] E. W. Myers, "Whole-genome DNA sequencing," *Computing in Science ' Engineering*, vol. 1, no. 3, pp. 33–43, 1999.

[5] J. L. Weber and E. W. Myers, "Human whole-genome shotgun sequencing," *Genome Research*, vol. 7, no. 5, pp. 401–409, 1997.

[6] V. Magrini, W. C. Warren, J. Wallis, et al., "Fosmid-based physical mapping of the *Histoplasma capsulatum* genome," *Genome Research*, vol. 14, no. 8, pp. 1603–1609, 2004.

[7] W. J. Kent and D. Haussler, "Assembly of the working draft of the human genome with GigAssembler," *Genome Research*, vol. 11, no. 9, pp. 1541–1548, 2001.

[8] D. H. Huson, K. Reinert, and E. W. Myers, "The greedy path-merging algorithm for contig scaffolding," *Journal of the ACM*, vol. 49, no. 5, pp. 603–615, 2002.

[9] D. Gordon, C. Abajian, and P. Green, "Consed: a graphical tool for sequence finishing," *Genome Research*, vol. 8, no. 3, pp. 195–202, 1998.

[10] D. Gordon, C. Desmarais, and P. Green, "Automated finishing with autofinish," *Genome Research*, vol. 11, no. 4, pp. 614–625, 2001.

[11] "*Phrap*," . http://www.genome.washington.edu/UWGC/ analysis tools/Phrap/htm.

[12] M. Pop, D. S. Kosack, and S. L. Salzberg, "Hierarchical scaffolding with Bambus," *Genome Research*, vol. 14, no. 1, pp. 149–159, 2004.

[13] S. H. Hong, J. S. Kim, S. Y. Lee, et al., "The genome sequence of the capnophilic rumen bacterium *Mannheimia succiniciproducens*," *Nature Biotechnology*, vol. 22, no. 10, pp. 1275–1281, 2004.