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.

Copyright © 2008 Pan-Gyu Kim et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

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
the features needed for general scaffold analysis and visualization have not been provided. 

GigAssembler can orient the contigs based on mRNA, paired plasmid ends, EST, and BAC end pairs. 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 kbp (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 C1 contig and b read are in the same direction, whereas g has direction C because the C2 contig and g read are in opposite directions. The size of the mate-pair helps to estimate the gap size between contigs C1 and C2. 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}, ..., c_{n}\}, a mate-pair set M = \{m_{1}, m_{2}, m_{3}, ..., m_{l}\}, and a set of reads R = \{r_{1}, r_{2}, r_{3}, ..., r_{k}\}, let G denote the scaffold graph using C and M:

\[ G = (C, E). \]  (1)

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_{l}(b) and P_{r}(g) be the start and end positions of b read in C_{1}, respectively, and let P_{l}(g) and P_{r}(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*.

<table>
<thead>
<tr>
<th>Genome</th>
<th>Genome length</th>
<th>Fold</th>
<th>Number of reads</th>
<th>Number of mate-pairs</th>
<th>Proportion of mate-pair reads relative to number of reads</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>M. succiniciproducens</em></td>
<td>2.3 Mbp</td>
<td>13.2</td>
<td>about 25,000*</td>
<td>275</td>
<td>2.2%</td>
</tr>
<tr>
<td><em>V. vulnificus</em></td>
<td>5.1 Mbp</td>
<td>11.7</td>
<td>76,971</td>
<td>1,781</td>
<td>4.5%</td>
</tr>
</tbody>
</table>

* 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.

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

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

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

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 \( \text{mate-pair size} - \{(C_1 \cdot \text{length} - P_e(b)) + P_e(g)\} \).

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

\[ \text{b-U and g-U: Gap} \ (C_1, C_2) = \text{mate-pair size} - \{(C_1 \cdot \text{length} - P_e(b)) + (C_2 \cdot \text{length} - P_e(g))\} \]
\[ \text{b-U and g-C: Gap} \ (C_1, C_2) = \text{mate-pair size} - \{(C_1 \cdot \text{length} - P_e(b)) + P_e(g)\} \]

\[ \text{b-C and g-U: Gap} \ (C_1, C_2) = \text{mate-pair size} - \{P_e(b) + (C_2 \cdot \text{length} - P_e(g))\} \]
\[ \text{b-C and g-C: Gap} \ (C_1, C_2) = \text{mate-pair size} - \{P_e(b) + P_e(g)\} \]

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.

<table>
<thead>
<tr>
<th>Data name</th>
<th>Errors*</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>MH1</td>
<td>Self Collision</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Gap size</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Overlap</td>
<td>22</td>
<td>2</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>MH2</td>
<td>Self collision</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Gap size</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Overlap</td>
<td>20</td>
<td>2</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>MH3</td>
<td>Self collision</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Gap size</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Overlap</td>
<td>18</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>MH4</td>
<td>Self collision</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Gap size</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Overlap</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>VV</td>
<td>Gap size</td>
<td>16</td>
<td>16</td>
<td>10</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>Overlap</td>
<td>65</td>
<td>7</td>
<td>3</td>
<td>2</td>
</tr>
</tbody>
</table>

* 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 $C_1(C_1 \cdot \text{length} - P_r(b))$ and the length of part of the mate-pair library in contig $C_2(P_r(g))$ are calculated. Finally, the gap size is calculated as

$$\text{mate-pair size} = \{|C_1 \cdot \text{length} - P_r(b)| + P_r(g)| \}.$$  

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.

<table>
<thead>
<tr>
<th>Comparison item</th>
<th>ConPath</th>
<th>Consed</th>
<th>Autofinish</th>
<th>Bambus</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy of scaffold</td>
<td>Medium</td>
<td>Medium</td>
<td>Medium</td>
<td>Strong</td>
</tr>
<tr>
<td>Construction time</td>
<td>Strong</td>
<td>Strong</td>
<td>Strong</td>
<td>Strong</td>
</tr>
<tr>
<td>Visualization</td>
<td>Strong</td>
<td>Medium</td>
<td>Weak</td>
<td>Weak</td>
</tr>
<tr>
<td>Error detection</td>
<td>Strong</td>
<td>Medium</td>
<td>Medium</td>
<td>Medium</td>
</tr>
<tr>
<td>Additional information</td>
<td>Strong</td>
<td>Strong</td>
<td>Medium</td>
<td>Medium</td>
</tr>
</tbody>
</table>

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.

ConPath 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 DISCUSSION

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 1vspace1pt 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

Submit your manuscripts at
http://www.hindawi.com