International Journal of Biological, Life and Agricultural Sciences
ISSN: 2415-6612
Vol:6, No:7, 2012

SIMGraph: Simplifying contig graph to improve *de novo* genome Assembly using next-generation sequencing data

Chien-Ju Li, Chun-Hui Yu, Chi-Chuan Hwang, Tsunglin Liu , Darby Tien-Hao Chang

Abstract*—De novo* genome assembly is always fragmented. Assembly fragmentation is more serious using the popular next generation sequencing (NGS) data because NGS sequences are shorter than the traditional Sanger sequences. As the data throughput of NGS is high, the fragmentations in assemblies are usually not the result of missing data. On the contrary, the assembled sequences, called contigs, are often connected to more than one other contigs in a complicated manner, leading to the fragmentations. False connections in such complicated connections between contigs, named a contig graph, are inevitable because of repeats and sequencing/assembly errors. Simplifying a contig graph by removing false connections directly improves genome assembly. In this work, we have developed a tool, SIMGraph, to resolve ambiguous connections between contigs using NGS data. Applying SIMGraph to the assembly of a fungus and a fish genome, we resolved 27.6% and 60.3% ambiguous contig connections, respectively. These results can reduce the experimental efforts in resolving contig connections.

***Keywords*—**contig graph, NGS, *de novo* assembly, scaffold

I. Introduction

GENOME sequencing and assembly are essential for understanding the genomes of organisms. Currently, next-generation sequencing (NGS) technologies, such as Roche 454 pyrosequencing[1], Illumina Genome Analyzer[2] and ABI SOLiD system[3], are prevailing due to their low cost and high throughput. It is now a common practice to obtain a deep coverage of sequences (also called reads) from a whole genome with one or a few NGS runs for assembly. However, genome assembly is still highly challenging. None of current programs can process sequencing reads into one single piece of DNA in one shot even for a small microbial genome of a few mega-bases. The resulting assembly usually appears as a set of long DNA fragments, called contigs.

A major challenge of *de novo* genome assembly arises because of the presence of repetitive DNA segments, called repeats, in genomes. When reads come from distinct copies of a repeat, assembly program usually cannot distinguish betweenthe reads by their genomic locations.

Chien-Ju Li is with Department of Electrical Engineering, National Cheng Kung University, Tainan 70101, Taiwan (email: n26981713@mail.ncku.edu.tw)
Chun-Hui Yu is withDepartment of Engineering Science, National Cheng Kung University, Tainan 70101, Taiwan (email:itsjeffreyy76@gmail.com)
Chi-Chuan Hwang is withDepartment of Engineering Science, National Cheng Kung University, Tainan 70101, Taiwan (email:chchwang@mail.ncku.edu.tw)
Tsunglin Liu is withInstitute of Bioinformatics and Biosignal Transduction, National Cheng Kung University, Tainan 70101, Taiwan (email:tsunglin@mail.ncku.edu.tw)
Darby Tien-Hao Chang is with Department of Electrical Engineering, National Cheng Kung University, Tainan 70101, Taiwan (email:darby@mail.ncku.edu.tw)

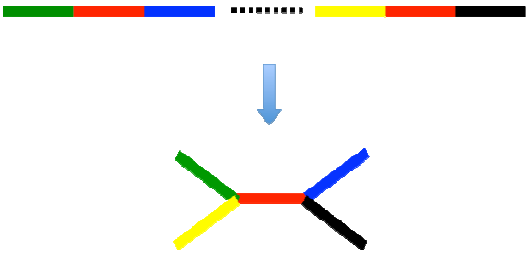
Thus, the reads from a repeat are often assembled into one DNA fragment (Figure 1a) with distinct flanking DNA connected, resulting in complicated connections between contigs, together called a contig graph (Figure 1).

Repeat problems in genome assembly can be serious for two reasons. First, repeats can constitute a significant portion of a genome. For example, DNA repeats occupy about half of the human genome[4]. Second, NGS reads are shorter (~400 bpfor 454, 100-150 bp for Illumina, and 75 bp for SOLiD) than traditional Sanger reads (800-1000 bp). A DNA repeat is actually not a repeat when the reads are longer than the repeat because with the unique part of reads outside the repeat, the reads can be distinguished. When reads are shorter, more DNA repeats appear. Even for a small microbial genome, repeat problems often result in a complicated contig graph (Figure 1b).

In this work, we propose a computational tool, SIMGraph, to simplify a contig graph for improving genome assembly. We note that with a deep read coverage, the assembled contigs cannot be connected mainly because of ambiguous connections instead of missing data[5]. Ambiguous connections arise when one contig connects to more than one other contig and the extra connections are false because of repeats and sequencing/assembly errors. SIMGraph simplifies a contig graph by resolving ambiguous contig connections, i.e., removing the false connections between contigs. After removing the false connections, some contigs can be re-connected unambiguously to improve the assembly.

SIMGraph takes advantage of two types of NGS data, 454 and Illumna paired-end (PE), to simplify a contig graph. More specifically, it uses Illumina PE data to resolve some ambiguous connections between contigs in a contig graph obtained with 454 data alone. Because 454 reads are longer than Illumina reads, we expect fewer repeats in the assembly with 454 data alone. In contrast, an Illumina platform yields a much greater amount of data than a 454 platform, thus providing a stronger statistical power for resolving ambiguities in contig connections.

(a)



(b)

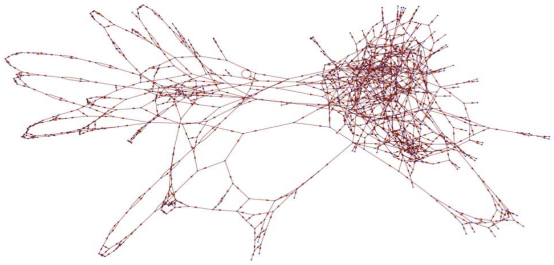


Fig. 1 *De novo* genome assembly with the presence of a repeat.(a)Shown on top are two genomic loci where each color represents a unique contig. The bottom graph is part of a so-called contig graph, which stores the information of all connections between contigs. The presence of repeats (segments in red) leads to branches of the contig graph. (b) A contig graph of the Velvet assembly of the Illumina PE reads simulated from the E. coli K12 MG1655 genome (~4.6 Mb). Each line represents a contig and the arrow indicates the direction. Only the largest connected group of contigs is shown here

Applying SIMGraph on the 454 and Illumina data of a fungus and a fish genome, we detected 666 and 3,708 ambiguous connections between contigs. SIMGraph then resolved 184 (27.6%), and 2,236 (60.3%) of the ambiguous connections. Though SIMGraph was tested using 454 and Illumina PE data in this work, it can be applied on Illumina PE data alone.

II. Method

A. NGS data and draft genome

We applied our tool on the NGS data of the fish, *Gasterosteusaculeatus*, in this study. The draft genome of the fish (BROAD S1, Feb 2006) has been released byBROAD Institute. This draft genome was assembled from Sanger sequencing data from several mate-pair libraries using Arachne2[6, 7]. The same fish sample was later sequenced again on 454 and Illumina platforms. The draft genome is 461,533,448 bpin length(in 21 chromosomes, 1 mitochondrial DNA, and 1822 scaffolds). We downloaded the draft genome from Ensembl[8]. We obtained the NGS data of the fish genome (sample ID SRS010092, Table I) from NCBI Sequence Read Archive (SRA) database[9]. We downloaded all the 454 data of the fish (a total of 3.7G bases in 11M reads) and the Illumina PE libraries (a total of 19G bases in 125M PEs of 76 bp read length), constituting an ~8X and ~41X coverage of the genome, respectively.

We also tested SIMGraph on the NGS data of a fungus, which was kindly provided by our collaborators. This dataset contains ~4.2M 454 reads and ~7M Illumina PEs, constituting an ~43X and ~16X coverage of the genome, respectively. The genome of this fungus has not been published.

B. Initial assembly of 454 data

We assembled the 454 data of the two genomes with Newbler[1]with default parameters. The resulting contig seqeuences (called 454 contigs) and contig graph, in the files 454AllContigs.fna and 454ContigGraph.txt, were submittedto SIMGraph for assembly improvement.

C. SIMGraph algorithm

Figure 2a shows the workflow of SIMGraph. SIMGraph takes as input the contig sequences and contig graph assembled with 454 data, and Illumina PE reads. After the following steps, it outputs the validity judgments of the detected ambiguous contig connections.

SIMGraph first detects a specific type of ambiguous connections between 454 contigs, named triads, from the contig graph (Figure 2a). A triad composes of three contigs forming two possible paths, C1-C3 and C1-C2-C3. In a triad, contigs C1 and C3 can either be connected straightly or connected with the contig C2 in the middle. This arises either because both theconnections exist in the genome, or because one of the two paths is false and appears due to sequencing or assembly errors. SIMGraph judges the validity of the two cases using Illumina PE data. The pseudo-code of the triad detection algorithm in SIMGraph is shown in Figure 2b.

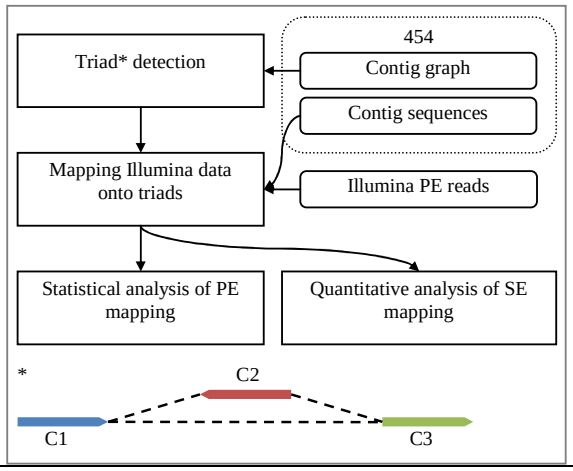
SIMGraph then maps the Illumina PE reads onto the contigsin the detected triads using SOAP2[10]. The mappings of Illumina PEs are classified into two categories: regular and bridging (Figure 3). A regular PE has its two reads mapped onthe different strands of the same contig (Figure 3a). From themappings of regular PEs, SIMGraph calculates the distributionof the distances between two paired reads. This distribution islater used to judge the validity of contig connections. A bridgingPE has its two reads mapped on different contigs (Figure 3b).SIMGraph uses the bridging PEs whose two reads are mapped on the contigs C1 and C3 of triads for resolving ambiguities in contig connections.

TABLE I
STATISTICS OF THE NGS DATA USED IN THIS STUDY

Species	No. of 454 reads (bases)	No. of Illumina read pairs (bases)
G. aculeatus[1]	11,109,932 (3,730,459,022)	125,373,070 (19,059,697,520)
Our fungus[2]	4,203,993 (1,413,313,543)	6,998,197 (531,862,972)

[1]The 454 libraries were found by searching SRA using the sample ID SRS010092 and the keyword "454".The read length and insert length of the Illumina libraries are 76 and 410 bp, respectively.[2]Illumina read length is 38 bp

(a)



(b)

For each contig *a* {

For each contig *b* that connects to *a* {

Suppose that the ends (5' or 3') of *a* and *b* in this connection are *a.e* and *b.e*, respectively.For each contig *c* that connects to *a.e* {

If *b.e* connects to the other end of*c*, then *a*, *c* and *b* are reported as the C1, C2 and C3, respectively of a triad. Note that C1 and C3 in a triad are interchangeable by the definition. Thus if *b*, *c* and *a* have been reported as a triad, than this triad will be ignored.

}

}

}

Fig. 2 Workflow and pseudo-codes of SIMGraph. (a) SIMGraph detects triads in the 454 assembly, performs read mapping and analyzes the mapping results with two methods. The 454 contig graph that describes connections between contigs is in '454ContigGraph.txt' of Newbler's outputs; while the 454 contig sequences are in '454AllContigs.fna'. This study defines a triad as a specific type of ambiguous gap. C1, C2 and C3 are 454 contigs, where the sharp ends indicate the 3'-ends. The dashed lines among them indicate their connections reported by Newbler. (b) Pseudo-codes of triad detection algorithm in SIMGraph

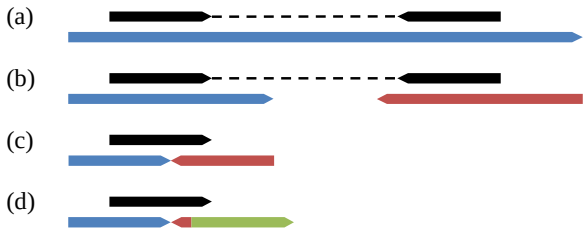


Fig. 3 Four conditions of mapping Illumina reads onto 454 contigs. Black lines are Illumina reads while color lines are 454 contigs. The dashed lines in between the two Illumina reads indicate that they are paired ends. (a) A Illumina PE maps on the same 454 contig. (b) A Illumina PE maps on two 454 contigs, denoted bridging PE in the context. (c) A Illumina SE crosses the junction of two 454 contigs. (d) A Illumina SE crosses the two junctions of three 454 contigs

For each triad, SIMGraph judges the validity of the two contig paths using a statistical analysis and a quantitative analysis. The statistical analysis focuses on the distancesbetween two paired reads, the paired-end distances (PEDs), of Illumina data. From the paired reads mapped on C1 and C3 of a triad, SIMGraph calculates two PEDs, $d_{1\text{-}3}$ and $d_{1\text{-}2\text{-}3}$, for the two paths C1-C3 and C1-C2-C3, respectively (Figure 4a). When a triad has *n* bridging PEs with *n* larger than a predefined parameter *b*, SIMGraph obtains the two PED distributions of $d_{1\text{-}3}$ and $d_{1\text{-}2\text{-}3}$, and then uses the Kolmogorov-Smirnov test (KS-test) to compare them with the PED distribution of the regular PEs. When $n<b$, SIMGraph calculates the two geometric means of probability densities—usually named likelihood—of the $nd_{1\text{-}3}$ and $nd_{1\text{-}2\text{-}3}$, respectively, using the probability distribution function in PED distance of the regular PEs. When comparing the PED of a path with the regular case, the larger p-value of KS-test or likelihood indicates that the corresponding path is supported by Illumina PE data. Specifically, we set a p-value cutoff (default 0.001, adjustable), above which the path is considered accepted by Illumina PE data. Figure 4 shows a sample result of SIMGraph's statistical analysis.

In the quantitative analysis, SIMGraph treats the mappings of Illumina data as of single reads without considering the pairing information. Briefly, we obtained the number of Illumina single reads that are mapped at the junction of contig connections. We define the support of the C1-C3 path as the number of single reads spanning the junction of C1 and C3 (Figure 3c). The support of the C1-C2-C3 path is defined as the smaller support of C1-C2 and C2-C3. If C2 is shorter than a predefined parameter o, the support of the C1-C2-C3 path is defined as the number of mapped single reads spanning C1, C2 and C3 (Figure 3d). If C1 and/or C3 are shorter than o, SIMGraph extends their outer ends (ends not connected to C2) by the corresponding 454 raw reads to enable read mapping. The inner ends (ends connected to C2) of C1 and C3 remain intact during contig extension. This may result in multiple extended C1 and C3 since a 454 contig is usually the consensus of multiple 454 raw reads. In this condition, the sum of supports of all the extended contigs is used. We set a support cutoff, above which we consider the path as accepted by Illumina PE data. Figure 5 shows a sample result of SIMGraph's quantitative analysis.

In the final output, we combine the results of statistical analysis and quantitative analysis. For each triad, when the two inferences agree, the result is strongly supported. If one of the analyses does not accept any of the paths, we consider the analysis non-informative and use the inference of the other analysis as the final inference. Such non-informative cases often arise because only few PEs or reads support the path. When both analyses are non-informative, the triad is considered non-resolvable. In contrast, if one of the analyses accepts both paths, indicating that the information content is enough, we expect that the other analysis also accepts both paths. If the other analysis does not accept both paths, we tend to be conservative and use the inference of the one path as the final inference.

This strategy is reasonable. For example, when the length of C2 is small, the statistical analysis tends to accept both paths because it cannot distinguish the two paths. It is then better to determine the final inference based on the quantitative analysis. When each of the two analyses infers only one path and the two inferred paths disagree, we consider the case inconsistent.

III. CABOG and SSPACE

We compared the performance of SIMGraph with two other programs, CABOG[11, 12] and SSPACE[13]. We emphasize that the two programs are not designed to simplify contig graph, but they contain algorithms that connect contigs. Because the module for contig connections is embedded in CABOG and cannot be run separately, we ran CABOG with the 454 and Illumina PE data and used the final assembly for performance comparison. SSPACE is a scaffolder program, and sometimes gives the sequences between contigs on a scaffold.

IV. Results and Discussions

A. Initial 454 assembly

We used Newbler to assemble the 454 reads of the fungus into 20,949 contigs. About half (10,486) of the contigs are at least of length 100 bp, in which the total number of bases is 32,828,399. The assembly outputs 31,157 connections between 20,919 contigs. Thus, almost all contigs are involved in the contig graph, suggesting a good coverage of the 454 data because. However, for a genome of size about 40 Mb, this initial assembly is much more fragmented in our experience. We later realized that the fungus sample is diploid, which reasonably explains the abundance of small contigs.

For the assembly of the fish genome, we obtained 235,498 contigs. The majority of the contigs (193,423) are at least of length 100 bps, in which the total number of bases is 407,226,149. The assembly outputs 147,106 connections between 130,182 contigs. That is, a significant portion of the contigs is not in the contig graph, suggesting that the 8X coverage of the 454 data is barely enough.

B. Mappings of Illumina PE reads

Using the contigs assembled from 454 data as a reference, we found that the majority of the Illumina PE reads could be mapped onto the contigs. In the case of fish, 16,367,388 of the 125,373,010 Illumina reads could be mapped onto the contigs.From these mappings, we found 16,347,307 regular PEs and 20,081 bridging PEs. The PED distribution of the regular PEs peaked at 291 bp (Figure 4). This distribution of PED is a very accurate source of information for our statistical analysis. We note that the PED of the downloaded Illumina library is denoted to be about 410 bp, which is quite far away from our calculated peak value of distribution. Our self-derived PED distribution is thus an advantage. In the case of fungus, 5,407,881 of the6,998,179Illumina reads are mapped onto the contigs.

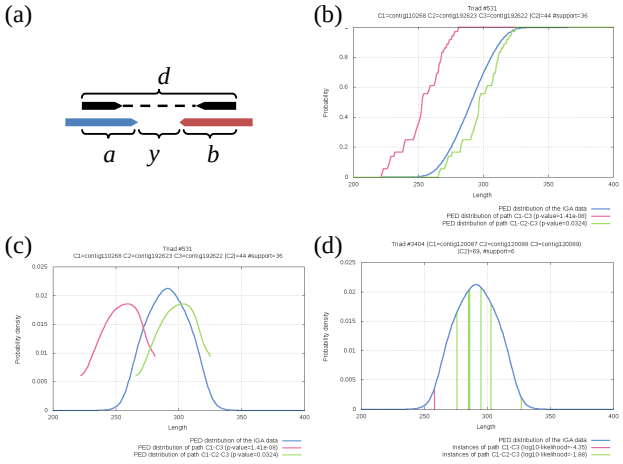


Fig. 4 Statistical analysis of SIMGraph. (a) Scheme of determination of paired-end distance (PED) in SIMGraph. Using SOAP2, two paired reads (the two black lines connected with a dashed line) are mapped on two contigs C1 and C3 (blue and red lines). SOAP2 outputs the mapped positions of the paired reads on the contigs, thus *a* and *b* are known constants. Since the gap sequence between C1 and C3 is either an empty sequence or the C2 contig, the PED is either $d_{1\text{-}3}=a+b$ or $d_{1\text{-}2\text{-}3}=a+y+b$ where *y* is the length of C2 contig. (b) This analysis uses Kolmogorov-Smirnov test (KS-test) to compare the cumulative distribution functions (c.d.f) of the PEDs of C1-C3 path (i.e., $d_{1\text{-}3}$, red line) and of C1-C2-C3 path (i.e., $d_{1\text{-}2\text{-}3}$, green line) to the background distribution (blue line). The p-values of KS-test are shown in the legends of the lines. (c) SIMGraph also provides the probability density functions (p.d.f) of PED. (d) If a triad has too few bridging PEs, SIMGraph resorts to likelihoods of the PEDs of C1-C3 path (red lines) and of C1-C2-C3 paths (green lines) belonging in the background distribution (blue line). The likelihoods are shown in the legends of the lines

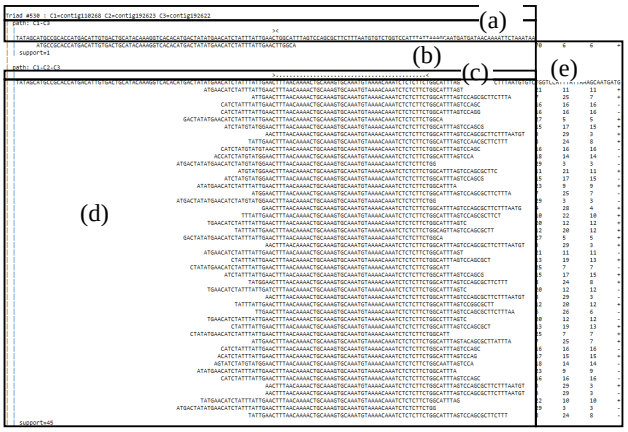


Fig. 5 Result of the quantitative analysis of SIMGraph. This analysis reports the number of single-end (SE) reads spanning paths C1-C3 and C1-C2-C3. (a) The sequence of the path C1-C3, where C1 and C3 are connected and'>' and '<' indicate the boundary of C1 and C3, respectively. (b) Supported SE reads of path C1-C3 and their alignments to the sequence connecting C1 and C3. (c) The sequence of the path C1-C2-C3, where '.' indicate C2. (d) Supported reads and alignments of path C1-C2-C3. (e) Details of each supported SE read, including the overlap with C1, the overlap with C3, the smaller overlap of the previous two and the direction of the SE read. A valid alignment requires that the smaller overlap exceeds *o* nucleotides and at most 5% mismatch in the whole alignment

C. SIMGraph

From the contig graphs of the fungus and fish assembly, SIMGraph detected 666 and 3,708 triads, and resolved 184 (27.6%) and 2,236 (60.3%) triads, respectively (Table 2). In these inferences, we found that the statistical and quantitative analyses were quite consistent.

For the fungus and the fish assembly, only in four (4/184=2.2%) and 49 (49/2236=2.2%) triads the two analyses were inconsistent.

Among the 2,236 inferences in the case of fish, 1849, 387, and 72 triads were inferred to go with the C1-C3, C1-C2-C3, and both paths, respectively (Table 2). To investigate the consistency between the SIMGraph inferences and the fish genome draft, we aligned the 454 contigs to the genome draft using BLAT[14]and checked which of the inferences is supported on the genome draft. Specifically, we aligned the C1 and C3 contigs of each triad, and check whether there is a DNA segment of length C2 in-between the two mapping loci of C1 and C3. Among our 1849, 387, and 72 inferences, 742, 149, and 5 inferences were consistent with the case on the fish genome draft. We note that these numbers do not directly imply a low performance of SIMGraph. It is also possible that SIMGraph points out some parts in the fish genome draft that deserve more careful inspections because SIMGraph provides detailed statistical and quantitative analyses. In addition, because the coverage of NGS reads is much higher than that of the traditional Sanger reads for the fish genome, it is not surprising that NGS data explore the genome more thoroughly.

D.Performance comparision

We compared the performance of SIMGraph with two related programs: CABOG and SSPACE. The two programs are not designed to simplify contig graph, but in each program a module that connects contigs is contained. To stand on equal footage, we checked whether the three tools connect the contigs in the detected triads. We found that SIMGraph resolved more triads compared with CABOG and SSPACE (Table 2) in both cases of the fungus and the fish. When compared to the fish genome draft, SIMGraph also inferred the largest number of consistent inferences. Taken together, SIMGraph achieved the greatest number of resolved triads while sacrificing a bit the accuracy.

E. Speed and memory usage

The four steps in Figure 2 can be grouped into the core algorithm of SIMGraph (Triad detection, Statistical analysis of PE mapping, and Quantitative analysis of SE mapping) and the read mapping step by SOAP2. The main CPU loading is at the read mapping step by SOAP2. For example, the core algorithm took ~2 hours while SOAP2 took ~10 hours for the fish case. The computational time of SOAP2 is proportional to the genome size. As for the memory, both the core algorithm and SOAP2 consumed the size of the genome. Thus, machines with a 4 GB are able to handle a usual mammalian genome.

F. Other types of ambiguous contig connections

Currently, SIMGraph focuses on a specific type of graph structure. In fact, the design of SIMGraph allows the possible extensions to other types of graph structures. Specifically, using Illumina PE data, it is possible to explore all pairs of contigs bridged by PEs whatever the connections between the two contigs are. We have analyzed the contig graphs of the two adopted genomes. The fish genome has 147,106 junctions of which 7,936 (accounting for 5.4%) may be resolved by SIMGraph. Here junctions are possible connections between contigs, which were reported but not actually connected by the assembler. The fungus genome has 31,832 junctions of which 1,362 (accounting for 4.3%) may be resolved by SIMGraph.

Even with such a small percentage, SIMGraph still contributes in three aspects. First, SIMGraph connects contigs with full sequences instead of a stretch of N's, which were observed in some cases using SSPACE. Second, SIMGraph explicitly provides statistical and quantitative measures to validate its predictions (Figure 4 and 5). Third, SIMGraph is suitable as an extra step in an assembly pipeline. Currently there are only few stand-alone assembly improvement algorithms that are independent to the assembly pipeline.

V. Conclusion

SIMGraph combines 454 andIllumina data to improve a genome assembly via simplifying the 454 contig graph, i.e., resolving ambiguous connections between contigs. Comparedwith two related programs, SIMGraph achieved the largest number of resolved ambiguous contig connections while scarifying a bit the accuracy, thus reducing experimental effortsfor such resolutions. SIMGraph provides detailed statistical and quantitative analyses for resolving ambiguities and the two analyses can be extended to resolve other configurations of contig connections. Moreover, through the detailed data provided by SIMGraph, one can study the mechanisms of sequencing and assembly errors leading to the paths un-supported by Illuminadata. Thus, our tool shall be of interest to scientists in the field of genome assembly.

Acknowledgment

We thank Dr. Wen-Hsiung Li for providing us the fungus NGS data, and Dr. Arthur Chun-Chieh Shih for the idea of extending short contigs using 454 raw reads. This work was supported by National Science Council Taiwan (NSC 100-2221-E-006-259, 99-2628-E-006-017 and 99-2745-B-006-003). Conflict of Interest: none declared.

TABLE II
RESULTS OF SIMGRAPHON *G. ACULEATUS* AND A DRAFT FUNGUS GENOME SEQUENCED BY OUR COLLABORATORS. WE NOTE THAT ALTHOUGH THE NUMBER OF RESOLVED TRIADS CONSISTENT WITH THE GENOME DRAFT MAY NOT SEEM HIGH, IT DOES NOT IMPLY A LOW PERFORMANCE OF SIMGRAPH. ON THE CONTRARY, BECAUSE SIMGRAPH DOES DETAILED STATISTICAL ANALYSIS AND QUANTITATIVE ANALYSIS, IT IS ALSO POSSIBLE THAT THE GENOME DRAFT CAN BE IMPROVED FURTHER USING SIMGRAPH

Organism	#C1-C3[1]	#C1-C2-C3[2]	#both paths[3]	#no path[4]	#inconsistent[5]	Solved (%solved)[6]
G. aculeatus						
SIMGraph (consistent with genome draft)	1,849 (742)	387 (149)	72 (5)	1,351 (447)	49	2,236 (60.3%)
CABOG (consistent with genome draft)	514 (249)	591 (295)	21 (15)	2,582 (690)	N/A	1,105 (29.8%)
SSPACE (consistent with genome draft)	160 (86)	90 (52)	7 (7)	3,451 (832)	N/A	250 (6.7%)
Our fungus						
SIMGraph	130	54	49	429	4	184 (27.6%)
CABOG	6	5	0	655	N/A	11 (1.7%)
SSPACE	3	14	0	649	N/A	17 (2.6%)

This table shows the number of triads where [1]only C1-C3 was accepted, [2]only C1-C2-C3 was accepted, [3]both paths were accepted and [4]no path was accepted. [5]Number of triads where one analysis accepted only C1-C3 but the other accepted only C1-C2-C3. [6]Sum of C1-C3, C1-C2-C3 and the ratio to the total detected triads.

REFERENCES

[1] O. M. Margulies, et al., "Genome sequencing in microfabricated high-density picolitre reactors," Nature, vol. 437, pp. 376-80, Sep 15 2005.
[2] D. R. Bentley, "Whole-genome re-sequencing," Curr Opin Genet Dev, vol. 16, pp. 545-52, Dec 2006.
[3] A. Valouev, et al., "A high-resolution, nucleosome position map of C. elegans reveals a lack of universal sequence-dictated positioning," Genome Res, vol. 18, pp. 1051-63, Jul 2008.
[4] M. A. Batzer and P. L. Deininger, "Alu repeats and human genomic diversity," Nat Rev Genet, vol. 3, pp. 370-9, May 2002.
[5] N. Nagarajan, et al., "Finishing genomes with limited resources: lessons from an ensemble of microbial genomes," BMC Genomics, vol. 11, p. 242, 2010.
[6] D. B. Jaffe, et al., "Whole-genome sequence assembly for mammalian genomes: Arachne 2," Genome Res, vol. 13, pp. 91-6, Jan 2003.
[7] F. C. Jones, et al., "The genomic basis of adaptive evolution in threespine sticklebacks," Nature, vol. in press, 2012.
[8] P. Flicek, et al., "Ensembl 2011," Nucleic Acids Res, vol. 39, pp. D800-6, Jan 2011.
[9] E. W. Sayers, et al., "Database resources of the National Center for Biotechnology Information," Nucleic Acids Res, Dec 2 2011.
[10] R. Li, et al., "SOAP2: an improved ultrafast tool for short read alignment," Bioinformatics, vol. 25, pp. 1966-7, Aug 1 2009.
[11] J. R. Miller, et al., "Aggressive assembly of pyrosequencing reads with mates," Bioinformatics, vol. 24, pp. 2818-24, Dec 15 2008.
[12] E. W. Myers, et al., "A whole-genome assembly of Drosophila," Science, vol. 287, pp. 2196-204, Mar 24 2000.
[13] M. Boetzer, et al., "Scaffolding pre-assembled contigs using SSPACE," Bioinformatics, vol. 27, pp. 578-9, Feb 15 2011.
[14] W. J. Kent, "BLAT--the BLAST-like alignment tool," Genome Res, vol. 12, pp. 656-64, Apr 2002.