BIOINFORMATICS ORIGINAL PAPER Vol. 28 no. 11 2012, pages 1455–1462
doi:10.1093/bioinformatics/bts162

Genome analysis Advance Access publication April 11, 2012

A *de novo* metagenomic assembly program for shotgun DNA reads

Binbin Lai[1,2,3], Ruogu Ding[1,2], Yang Li[1,2], Liping Duan[4] and Huaiqiu Zhu[1,2,3,*]

[1]State Key Lab for Turbulence and Complex Systems and Department of Biomedical Engineering, College of Engineering, [2]Center for Theoretical Biology, [3]Center for Protein Science, Peking University, Beijing 100871 and [4]Department of Gastroenterology, Peking University Third Hospital, Beijing 100191, China

Associate Editor: Alex Bateman

ABSTRACT

Motivation: A high-quality assembly of reads generated from shotgun sequencing is a substantial step in metagenome projects. Although traditional assemblers have been employed in initial analysis of metagenomes, they cannot surmount the challenges created by the features of metagenomic data.

Result: We present a *de novo* assembly approach and its implementation named MAP (metagenomic assembly program). Based on an improved overlap/layout/consensus (OLC) strategy incorporated with several special algorithms, MAP uses the mate pair information, resulting in being more applicable to shotgun DNA reads (recommended as >200 bp) currently widely used in metagenome projects. Results of extensive tests on simulated data show that MAP can be superior to both Celera and Phrap for typical longer reads by Sanger sequencing, as well as has an evident advantage over Celera, Newbler and the newest Genovo, for typical shorter reads by 454 sequencing.

Availability and implementation: The source code of MAP is distributed as open source under the GNU GPL license, the MAP program and all simulated datasets can be freely available at http://bioinfo.ctb.pku.edu.cn/MAP/

Contact: hqzhu@pku.edu.cn

Supplementary information: Supplementary data are available at *Bioinformatics* online.

Received on October 12, 2011; revised on March 27, 2012; accepted on April 2, 2012

1 INTRODUCTION

Processing of shotgun metagenomic sequence data usually does not have a fixed end point to recover one or more complete genomes as for isolated microbial genomes, sometimes with the exception of finishable dominant populations (Kunin *et al*., 2008). Nevertheless, with the proliferation of metagenomic projects, the assembly tools, which aim to combine sequence reads into contiguous stretches of DNA called contigs, are still expected to play an important role in sequence processing, due to more valuable genomic content they can provide (Gill *et al*., 2006; Kunin *et al*., 2008; Tasse *et al*., 2010; Tyson *et al*., 2004; Venter *et al*., 2004). Recently, the comparative assembly approach such as AMOS comparative assembler, which uses a reference genome or closely related species to align reads, was applied to facilitate assembly of short reads (Kunin *et al*., 2008; Pop *et al.,* 2004). However, because of the potential bias caused by phylogenetic complexity and diversity, the *de novo* assembly methods are still regarded as irreplaceable tools for accurately assembling the novel genomic sequences that broadly exist in the metagenomic sequencing data (Pop, 2009; Pop *et al*., 2007).

As the high-throughput sequencing technologies by the next-generation sequencing platforms such as Illumina (http://www.illumina.com), SOLiD (http://www.appliedbiosystems.com) and Helicos (http://www.helicosbio.com) are available in metagenomic projects, many efforts have been devoted to develop assembly tools such as SSAKE (Warren *et al.*, 2007), Velvet (Zerbinor *et al.*, 2008) and EULER-SR/EULER-USR (Chaisson *et al*., 2009). However, these methods are not targeting the metagenome sequencing. Moreover, compared with Sanger and 454 sequencing, the current limitation of shorter reads (<200 bp, typically 25–100 bp) and higher errors by the new sequencing platforms does not allow a significant utility for metagenomic analyses for the difficulty in phylogenetic study or gene function inference (Miller *et al.*, 2010; Rodrigue *et al*., 2010; Wommack *et al*., 2008). In fact, shorter reads technologies have not been widely used in metagenome sequencing, and meanwhile the sequencing technologies producing longer reads, such as Sanger (usually 700–1000 bp) and 454 sequencing (usually 200–500 bp), are still the overwhelming recommendation and thus remain the major source of metagenomic sequence data (Wommack *et al*., 2008). Therefore, it is never trivial to continue to emphasize the importance of longer reads to metagenomic analyses, clearly including the reads assembly tool designed specifically.

To date, for such longer shotgun reads in many metagenome projects, the assembly still relies largely on the existing tools demonstrating high performance to manage for single genome, such as Phrap (http://www.phrap.org), Celera Assembler (Miller *et al.*, 2008; Myers *et al*., 2000) and PCAP (Huang *et al.*, 2003). However, the assembly of a community of genomes is different from the assembly of a single genome. In the assembly of single genomes, the fundamental problem is the presence of repeated DNA fragments in the target sequence that often leads to assembly errors. However, metagenomic assembly has its particular difficulties due to two challenges (Kunin *et al*., 2008): the genomic repeats may originate from either the same genome or the different genomes, and the inhomogeneous coverage distribution and the low abundance of organisms provide limited information to handle repeats. Unfortunately, the design for the task of single-genome assembly prevents the traditional *de novo* assembly software

*To whom correspondence should be addressed.

adapting well to metagenomic analysis. For example, most of genome-oriented assemblers have particular restrictions, such as uniform coverage, which is not suitable to metagenomic assembly. Thus, most of traditional assemblers demonstrated their performance with metagenomic data varies significantly from that with individual microbial genomes (Kunin *et al*., 2008). Studies have shown that the metagenome assembly by classic assemblers targeting individual genome project has enormous particular misassembled contigs named chimeras which consisted of reads from different genomes (Mavromatis *et al.*, 2007; Pignatelli *et al.*, 2011).

In this article, we focus on the metagenomic assembly problem of longer reads produced by Sanger (typically 700–1000 bp) and 454 sequencing (typically 200–500 bp). Meanwhile, mate pair information from both ends of a DNA fragment for a given size (e.g. an insert in a vector plasmid in Sanger sequencing or a mate pair template in 454 sequencing) in sequencing is introduced, which is commonly available in Sanger sequencing and most new sequencing technologies including 454 sequencing (Korbel *et al*., 2007; Metzker, 2010; Miller *et al.*, 2010). We sought to establish a new *de novo* approach called MAP (Metagenomic Assembly Program). The algorithm of MAP is designed based on an improved overlap/layout/consensus (OLC) strategy incorporated with several special algorithms. What is distinct about the algorithm is that we integrated the mate pair information into the layout stage of OLC strategy, resulting in being more applicable to metagenomic data and a higher performance of reads assembly. We assessed our method on simulated data compared with currently widely used assembly tools. Specifically, metagenomic reads were generated in length of averaged 800 bp and 200 bp, respectively, to meet the characteristics of the Sanger and 454 sequencing technology. We compared MAP with Phrap and Celera Assembler on 800 bp reads and with Celera Assembler, Newbler (Margulies *et al*., 2005) and Genovo (Laserson *et al*., 2011) on 200 bp reads. The de Brujin graph-based assemblers, which are more suitable for shorter reads ($<$100 bp), such as Velvet (Zerbinor *et al.*, 2008), SOAPdenovo (Li *et al*., 2010) and Meta-IDBA (Peng *et al*., 2011), are not included in the comparisons with MAP in this work, as they are less preferred when compared with the graph-based assemblers on longer reads ($>$100 bp) due to their limitations on longer reads (Schatz *et al*., 2010). The results show that for the number and size of assembled contigs, MAP presents a competitive assembly capacity compared with the other assemblers, whereas for the chimeric contigs and nucleotide sites match of assembled contigs, MAP demonstrates higher accuracies than the other assembly tools.

2 MATERIALS AND METHODS

2.1 Materials

To design the algorithm and benchmark program MAP and other assemblers, herein we used MetaSim to produce simulated metagenomic data. MetaSim is a widely used sequencing simulator to generate collections of synthetic reads reflecting the diverse taxonomical composition of typical metagenomic data (Richter *et al*., 2008).

Following Mavromatis *et al*. (2007), we constructed the simulated data of different community complexity (low, medium and high complexities, as LC, MC and HC, respectively), the LC dataset has only one organism dominating over others, the MC dataset has several dominant organisms and the HC dataset has no dominant organisms. From the microbial complete genomes at the NCBI repository (http://www.ncbi.nlm.nih.gov/sites/genome), we selected the same 113 species described by Mavromatis *et al*. (2007) to construct the simulated microbiomes (details are shown in Supplementary Figure S1 and Supplementary Table S1). If one special strain was not found, we chose a close relative (usually a different strain) following Pignatelli *et al*. (2011). Each metagenome was then used to generate two types of mate pair reads in average length to meet Sanger and 454 sequencing platforms (800 bp and 200 bp, respectively). For 800 bp datasets, we used MetaSim in Sanger model to generate error-free reads and the sequencing error reads with mean error rate 0.005 errors per base. The read length follows a normal distribution. For 200 bp datasets, we first used MetaSim in 454 model to generate the error-free mate pair reads with both paired end reads and the linker in one read. Each end of the pair is in exact length 200 bp. The mate pair reads were then processed by NGSfy tool [a tool used for generating 454 and Illumina reads, which has been used in Pignatelli *et al.* (2011)] to generate the 454 typical sequencing errors ($\sim$0.005 mean error rate). For both 800 bp and 200 bp datasets, mate pairs were generated with average length in 3 kb and SD in 200 bp (also following a normal distribution). Finally, we applied MetaSim and NGSfy to generate totally 12 simulated metagenome datasets with three different complexities and four read types.

All simulated datasets can be downloaded from the website http://bioinfo.ctb.pku.edu.cn/MAP/.

2.2 Methods

Before we describe the algorithm, it should be noted that our assembly method does not include the sequence trimming procedure (in terms of quality trimming and contaminant trimming) or the consideration to process the low-quality reads, therefore external software tools such as Lucy (Li *et al*., 2004) are recommended to perform such trimming before the data being input in this study.

2.2.1 Improved OLC strategy As mentioned in Introduction, metagenome assembly is greatly puzzled by the fragments mixture coming from different genomes, often leading to the so-called chimeric contigs. To address this problem, MAP designs an improved approach of the classical OLC strategy, in which several special algorithms are incorporated into its stages, to calculate correct contigs by connecting the fragments linked by mate pairs to prevent the false merge of unrelated reads.

The classical OLC strategy, which may well be the most popular assembly approach and have been used in Celera Assembler (Miller *et al*., 2008; Myers *et al*., 2000) and PCAP (Huang *et al*., 2003), consists of three stages. The first stage 'Overlap' is to calculate all-against-all pair-wised overlaps, based on which the reads are glued into contigs; the overlaps are then input into the second stage 'Layout', which builds an overlap graph of reads and their overlaps and then determines which reads and how they are arranged one by one (the relative position and orientation) and the third stage 'Consensus' goes to decide the DNA sequence implied from the reads arrangement from the layout stage.

For our improved OLC strategy, MAP deploys a series of algorithms in three stages as shown in Figure 1. In the overlap stage, the filter algorithm based on *q* gram (Mullikin *et al*., 2003) is used to obtain the read pairs that are supposed to have the overlaps, and the seed and extend alignment approach, similar to that used by BLAST (Altschul *et al*., 1990), is employed in the pairwise alignment calculation. More details are available in Supplementary Methods. In the consensus stage, a consistency-based consensus algorithm is used (Rausch *et al*., 2009), which is based on a multi-read alignment algorithm aligning the reads with a consistency-enhanced alignment graph of shared sequence segments identified in advance. The layout stage applying mate-paired information is presented in detail as below.

2.2.2 Algorithm in layout stage We now describe the algorithm in the layout stage integrated into mate-paired information to find the optimal paths to construct final contigs. The flowchart of the algorithm is presented in Figure 1 as well.

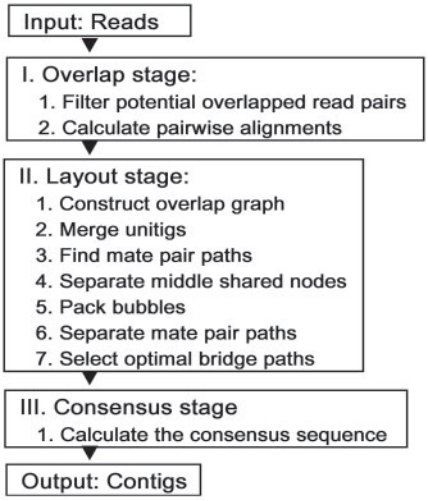

Fig. 1. The flowchart of MAP algorithm.

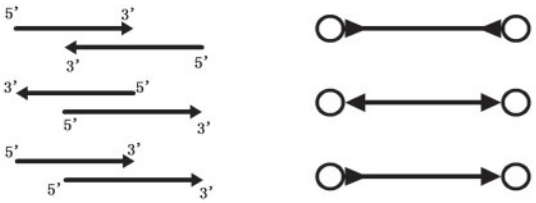

Fig. 2. Bi-directed edges in overlap graph stands by the overlaps of the reads. Each read can be in either of two orientations, whereas two of the cases (both to the left and both to the right) are symmetric.

In the current OLC approach, the overlap graph is used to facilitate the assembly process. Conceptually, reads and overlaps are represented in the graph G by nodes and bi-directed edges, respectively. The arrows of both ends of the edge are determined by the way how two reads overlap (Fig. 2). Herein, a dovetail path is defined as an acyclic path with each node has only one arrow outward it and one arrow inward it. Thus, a dovetail path can determine a certain contig by means of threading the reads corresponding to the nodes in this path. Thus, the goal of the layout stage is to separate the graph into disconnected dovetail paths. However, as there may be quite many misleading edges in the graph that represent the false overlaps mainly originated from two repetitive DNA regions or similar fragments of different genomes, this goal seems to be a formidable task. To this end, MAP is designed to determine the optimal dovetail paths as following steps:

(1) *Construct overlap graph*: Based on the output of the overlap stage, a bi-directed graph, named overlap graph, is constructed where the nodes and edges represent reads and overlaps, respectively. The overlaps where one read is completely contained in the other read are not included in graph construction.

(2) *Merge unitigs*: Unitig is designated to represent the contig in which the reads have no contested overlaps with any other reads (Myers *et al*., 2000). In overlap graph, a dovetail path without any other edges intersected can directly lead to a unitig, and the path is called a simple path. Herein, MAP uses the transitive reduction algorithm to remove the transitive edges (Myers, 2005). If three nodes A, B and C are linked by overlaps A-B, B-C and A-C, and the overlaps are mutually consistent among three reads, then the edge A-C is said as a transitive edge. Then a simple path may be replaced by a node representing the corresponding unitig, so that the graph can be reduced. Thus, the node in graph may be either a read or a set of reads in a simple path.

(3) *Find mate pair paths*: To further select the path from the graph with many conflicting edges, the algorithm then goes to find the paths with mate pair threading. As mate-paired reads come from the same longer DNA fragment, the path with mate pair threading can be regarded as an authentic path corresponding a true DNA fragment. A path is defined as a mate pair path, if both ends are linked by mate pairs, and distance of the mate reads in the path is consistent with the mate pair length (Fig. 3a, b). A description of the method of mate pair path finding can be referred to in Methods in Supplementary Materials. After the mate pair paths are decided in the graph, the edges that are not only excluded (i.e. not passed) in any mate pair paths but also intersect other mate pair paths are removed (Fig. 3c). After this step, many paths (probably not all) become simple paths.

There may be still mate pair paths that intersected each other. For example, multiple mate pair paths may cross each other and a pair of nodes may involve more than one mate pair path connecting them. In these cases, nodes in the graph are classified into stem nodes and bridge nodes, in which the bridge nodes are defined as not only being in the middle of the path but also having the length shorter than that of mate pairs, whereas the remaining nodes are classified as stem nodes.

(4) *Separate middle shared nodes*: If a bridge node is shared by more than one mate pair paths, we separate these mate pair paths by duplicating this bridge node for each mate pair path in the graph (Fig. 4a).

(5) *Pack bubbles*: If two stem nodes are linked by several paths including bridge nodes, thus forming a so-called bubble structure (Miller *et al*., 2010), MAP connects the stem nodes by replacing those paths (named bridge paths) with an altered node linking to the stem nodes, named bubble node, which does not represent a sequence but several sequences that link both the stem nodes (Fig. 4b).

(6) *Separate mate pair paths*: Steps 4 and 5 have separated different mate pair paths intersecting at the bridge nodes. However, some mate pair paths may still intersect at the stem nodes, such as the nodes corresponding to 'long repeats'. To separate these mate pair paths, a greedy approach is applied to always select the best mate pair path and cut edges that intersect the paths. Herein, 'best' is measured as the most mate pairs that thread the path. After a best mate pair path P_1 is marked, the next best mate pair path P_2 is found from the graph in the second round. As the edges intersecting P_1 have been removed, P_2

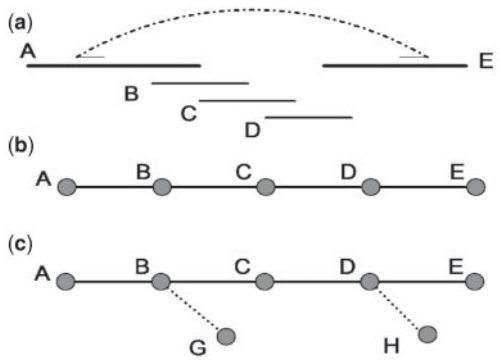

Fig. 3. (**a**) (**b**) illustrate an example of mate pair path ABCDE, with end nodes A and E having mate pairs linking. In (**c**), the edge BG and DH are not passed by any mate pair paths and they intersect the mate pair path ABCDE. Thus they are removed from the graph.

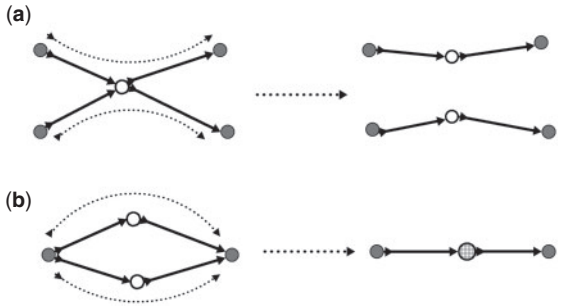

Fig. 4. (**a**) Separate the middle bridge node shared by two mate pair paths. (**b**) Replace the bulb structure with a simple path. The gray nodes represent stem nodes, the white nodes represent bridge nodes and the shadowed node represents bubble node.

does not intersect P_1, and the edges intersecting P_2 will be removed in the second round as before. Go on the next round until all the paths in the graph have been marked. Thus, the overlapping mate pair paths are merged, and the graph is finally separated into disconnected dovetail paths.

(7) *Select optimal bridge paths*: If a path has a bubble node that involves several bridge paths formed in the steps 4 and 5, optimal bridge path will be selected to replace the bubble node, so that all the nodes in the final path represent a DNA sequence. The rule is to select the one that has the most mate pairs linking nodes in the inner bridge and outer stem.

3 RESULTS

We compare the performance of MAP to Celera Assembler (Miller *et al*., 2008; Myers *et al*., 2000) and Phrap (http://www.phrap.org) on 800 bp simulated datasets, whereas compare to Celera Assembler and Newbler (Margulies *et al*., 2005) on 200 bp simulated datasets. Celera Assembler and Phrap are commonly used on Sanger reads and represent two styles of approaches, OLC approach and greedy approach. Newbler is the widely used software for 454 reads assembly distributed by 454 Life Sciences. The Celera assembler, which first targeted Sanger reads, distributed a new version for 454 reads assembly (Miller *et al*., 2008) and claimed to outperform Newbler. Newbler also uses OLC approach. Parameters were chosen to reduce the assembly chimerism. Specially, for error-free test sets, the minimal overlap identities for MAP, Celera and Newbler were set as 99.5%, and for error sequencing test sets, the minimal overlap identities were set as 98%. Detailed parameter settings are available in Supplementary Materials.

We conducted the comparison of different assemblers on simulated data by assessing the quality of contigs produced by them. Five independent quantities, defined to illustrate the assembly performance, are summarized in Tables 1 and 2. Specifically, we used the proportion of the reads assembled into contigs, the number of contigs and their average length (i.e. contig size) to evaluate the assembly capacity. The two measurements are used to evaluate the assembly accuracies: the proportion of chimeric contigs in total contigs and the proportion of the so-called divergent contigs in total contigs. Following the studies in Mavromatis *et al*. (2007) and

Table 1. The assembly results on simulated Sanger reads (800 bp)

	Assembled reads (%)	No. of contigs	Contig size (bp)	Len at 5 Mb (bp)	Chimeric contigs (%)	Divergent contigs (%)	Severe chimeras (%)
HC (error-free type)							
MAP	38.2	21 670	1288	1631	3.7	0.4	0.06
Celera	38.0	21 498	1283	1612	3.8	0.5	0.08
Phrap	41.6	22 222	1286	1598	10.3	6.7	0.46
MC (error-free type)							
MAP	65.3	12 664	2306	10 453	1.8	0.4	0.06
Celera	65.1	12 483	2331	11 961	1.9	0.5	0.07
Phrap	66.3	13 296	2214	10 038	7.7	5.0	0.70
LC (error-free type)							
MAP	55.7	10 134	1910	9856	2.4	0.4	0.02
Celera	55.5	10 040	1908	10 088	2.4	0.4	0.01
Phrap	57.4	10 802	1836	5849	11.1	7.3	0.50
HC (sequencing error type)							
MAP	39.1	21 988	1296	1623	6.2	0.9	0.40
Celera	38.5	21 062	1270	1599	8.5	1.0	0.45
Phrap	40.6	22 180	1265	1574	12.2	3.4	0.92
MC (sequencing error type)							
MAP	65.5	12 664	2301	10 702	5.6	1.1	0.56
Celera	65.4	11 901	2314	11 762	6.9	1.1	0.66
Phrap	65.8	13 930	2125	8396	9.6	2.4	1.12
LC (sequencing error type)							
MAP	55.8	10 269	1913	6586	5.2	0.9	0.44
Celera	55.7	9950	1906	7 856	7.9	1.1	0.63
Phrap	56.4	11 481	1770	4618	11.2	2.5	0.98

HC, MC and LC represent error-free datasets in high, medium and low complexities. Len at 5 Mb: contigs are sorted in decreasing order by length, and the number and size of the smallest contig to reach 5 Mb are counted. Divergent contigs are those have identity $< t$ ($t = 99\%$ for error-free type datasets, and $t = 95\%$ for sequencing error-type data), with respect to the reference genomes.

Table 2. The assembly results on simulated 454 mate pair reads (200 bp)

	Assembled reads (%)	No. of contigs	Contig size (bp)	Len at 1 Mb (bp)	Chimeric contigs (%)	Divergent contigs (%)	Severe chimeras (%)
HC (error-free type)							
MAP	34.3	78 732	300	854	4.4	0.4	0.04
Celera	33.5	76 960	298	475	4.5	0.5	0.07
Newbler	5.1	3409	454	312	17.1	2.7	0.29
MC (error-free type)							
MAP	64.0	55 712	502	5543	2.7	0.1	0.04
Celera	62.9	52 936	510	4071	3.1	0.1	0.05
Newbler	44.3	16 341	812	2178	14.5	2.5	0.46
LC (error-free type)							
MAP	53.8	40 168	443	13 638	2.8	0.1	0.02
Celera	53.0	38 964	445	12 106	2.9	0.1	0.05
Newbler	33.2	4092	1271	5168	9.0	1.2	0.25
HC (sequencing error type)							
MAP	16.0	33 212	359	854	6.6	3.0	0.17
Celera	10.9	21 340	218	551	12.9	2.8	0.36
Newbler	3.2	2331	504	231	19.0	4.6	2.35
MC (sequencing error type)							
MAP	52.0	36 087	590	5973	4.2	1.9	0.39
Celera	51.1	33 540	500	5328	6.3	2.9	0.56
Newbler	44.5	16 678	771	2993	19.4	3.6	0.65
LC (sequencing error type)							
MAP	43.0	20 775	602	11 346	3.4	2.5	0.31
Celera	40.5	17 115	515	10 461	5.6	2.8	0.43
Newbler	33.4	4440	1358	4933	14.9	1.6	0.94

HC, MC and LC represent error-free datasets in high, medium and low complexities. Len at 1 Mb: contigs are sorted in decreasing order by length, and the number and size of the smallest contig to reach 1 Mb are counted. Divergent contigs are those that have identity $< t$ ($t = 99\%$ for error-free type datasets, and $t = 95\%$ for sequencing error-type data), with respect to the reference genomes.

Pignatelli *et al.* (2011), chimeric contigs are defined as those which comprised reads from different organisms. A divergent contig is defined as having an identity $< t$ ($t = 99\%$ for error-free data and $t = 95\%$ for sequencing error data) comparing against the original reference genomes from which the metagenomic reads are sampled.

The results of the first three quantities of assembly capacity, namely the percentage of reads assembled into contigs, the number of contigs and their average length, are presented as follows. For the 800 bp datasets, the comparison of MAP, Celera and Phrap is presented in Table 1. In general, the three assemblers have a similar level in terms of the assembly capacity. Although Phrap shows a little more reads assembled and more contigs than both MAP and Celera, most often it generates shorter contigs than other two assemblers do. In fact, the assembly capacity of each assembler is locked in a 'zero-sum' game, any more for the number of contigs usually means a shorter average length of contigs. For the 200 bp datasets, the results of MAP, Celera and Newbler are listed in Table 2. On the error-free datasets, MAP and Celera have a similar performance on the assembly capacity. While on the sequencing error-type datasets, MAP outperforms Celera with higher proportion of assembled reads, larger contig number and longer contig size. On all 200 bp datasets, Newbler assembles many fewer reads in contigs and generates fewer contigs compared with both MAP and Celera. For example, in the HC, error-free data, Newbler only assembled 5.1% of reads into contigs, whereas MAP assembled 34.3% of reads. When assembling the MC data, Newler increases the assembled reads to 44.3%, however, still <64.0% by MAP. In the LC datasets of both error free and sequencing error type, Newbler has the longest contig average length among three assemblers. However, when we calculate the larger contigs by each assembler, we found that MAP has more large contigs than Newbler. For example, on the MC data of error-free type, MAP generates 4546 large contigs with length longer than 1000 bp, whereas Newbler generates 4059 large contigs (see Supplementary Table S3). This indicates that the shorter contig average length of MAP than Newbler is just caused by the much more short contigs generated by MAP. Therefore in conclusion, MAP demonstrates its high assembly capacity to same level of Celera and Phrap on the 800 bp datasets, whereas to a great extent outperforms both Celera and Newbler on the 200 bp datasets.

Misassembly has certainly negative effects on the further analysis phases such as gene calling or translation initiation signals finding (Hu *et al.*, 2009; Wommack *et al.*, 2008; Zhu *et al.*, 2010). Thus for a metagenomic assembler, there is great demand to improve the quality of assembly. However, the existing well-known assemblers applied in metagenome projects were designed originally for individual genomes, which generate enormous misassembly especially chimeric contigs when processing metagenomic reads (Kunin *et al.*, 2008). Chimeric contigs are composed of reads from different species and, thus, become a critical measure of assembler utility. In the current test on simulated data, the proportion of chimeric contigs in total contigs assembled by different assemblers is given (Tables 1 and 2). For all six 800 bp datasets, MAP demonstrates overall lower chimeric contig ratio than both Celera and Phrap do, whereas Phrap has much more chimeric contigs

usually with double ratio for sequencing error type and more than 3-fold ratio for error-free type compared with MAP. Considering the same level of the assembly capacity (the proportion of reads assembled, and the number and the size of contigs) for the three assemblers, it is clear that MAP shows the highest assembly accuracy to avoid chimeric contigs. Moreover, for the 200 bp datasets, MAP has the lowest chimeric contig ratio among three assemblers, whereas Newbler shows much higher chimeric ratio. The MAPs superiority is more evident for data of sequencing error type. This suggests that our method can especially well apply to more complex case of assembly processing (i.e. reads with sequencing errors compared with error free and shorter reads compared with longer reads). To investigate the misassembly source of chimeric contigs, we further calculated the taxonomic lowest common ancestor (LCA) of reads in each chimeric contig. Resultantly, all assemblers display a similar pattern in the distribution (shown in Supplementary Figure S2). The overwhelming majority of them are composed of organisms belonging to the same species, genus or families. This is consistent with the argument that the reads from closely related genomes are more likely to be assembled together (Kunin *et al*., 2008).

We inspect the assembly accuracy according to nucleotide sites match of assembled contigs. As most of the chimeric contigs are caused from closely related genomes, it inevitable that part of coassembled contigs have extremely high similarity, even full identity, in sequence compared with authentic DNA fragments in one genome. In this case, high match of nucleotides for assembled contigs should be taken into account a positive contribution to metagenomic analysis. Therefore, we perform an examination on the identity between the consensus sequence of contigs and the original genomes from which the simulated metagenomic reads have been sampled. Also, the so-called divergent contig is defined to have an identity $<99\%$ comparing against the original reference genomes. To calculate the identity, we used BLAST (Altschul *et al*., 1990) and BLAT (Kent, 2002) to align the contig against the reference genomes and then calculated the highest identity as the proportion of the matched bases in the contig. Tables 1 and 2 list the proportion of the divergent contigs. More comprehensive results are reported in Supplementary Table S2. The results for the 800 bp test sets show that MAP has the least divergent contig ratio among the three assemblers, whereas Phrap produces the highest ratio. Especially for error-free test sets, Phrap has >10 times the divergent contig ratio of both MAP and Celera. Similarly, for the 200 bp test sets, MAP also has the least proportion of the divergent contigs compared with Celera and Newbler. In the HC datasets, Newbler always has the highest divergent contigs and extremely few contigs, suggesting that Newbler is not suitable for assembling HC metagenomes without dominant species. We also changed different criteria to define the significant divergence between the contigs and the conference sequence (Supplementary Table S2). Resultantly, the change of criterion of the degree of divergence does not affect the conclusion that MAP has the lowest proportion of the divergent contigs among all the assemblers.

As presented in Tables 1 and 2, in most cases, the divergent contig ratio is lower than the chimeric contig ratio. This evidence supports the conclusion that most chimeric contigs do not represent severe chimeric errors but rather co-assembled strains among closed related genomes. Thus, we further calculated the severe chimeric errors, which are defined as below. We inspected into the alignments results of the divergent contigs as defined before against the reference genomes. If one contig does not have the whole sequence mapping to a reference but has different parts that map against different reference genomes, we deem the misassembly of the reads from different organisms as the reason of the divergence between the contig sequence and the references, thus we call this contig a severe chimeric contig. As presented in Tables 1 and 2, the severe chimeric contig ratio is much lower than the chimeric contig ratio. The proportion of the severe chimeric contig in the total chimeric contigs ranges from $\sim$1% to 10%, where this ratio can be as low as 1% in error-free test sets and also can reach 10% in the error sequencing test sets. We further found that most severe chimeric contigs are short contigs (of length <1 kb for 454 reads and of length <3 kb for Sanger reads) (see Supplementary Table 3). In most cases, MAP has the lowest chimeric contig ratio.

We also evaluated the assembly contiguity for MAP and other assemblers. By sorting the contigs in decreasing order by length, we may count the number and the length of the smallest contig required to reach 5 Mb in the 800 bp datasets and 1 Mb in the 200 bp datasets. Herein, we eliminated the divergent contigs to avoid an assembler generating longer but likely being error contigs to produce superior contiguity. As presented in Tables 1 and 2, the contiguity of MAP is similar with Celera and better than Phrap on the 800 bp datasets and Newbler on the 200 bp datasets.

It should be noted that the *de novo* assembler Arachne (Jaffe *et al*., 2003), which was shown in Mavromatis *et al*. (2007) to have the fewest chimeric contigs, seems to have a low degree of contiguity on metagenome datasets in our test (see Supplementary Table S5). For example, on the LC and 800 bp error-free dataset, Arachne assembled 34.2% of reads in contigs and generated 1456 contigs and 367 large contigs (of length $\geq$3 kb), whereas the corresponding statistics of MAP are 55.7%, 10 134 and 462, respectively. The low degree of contiguity of Arachne may be mainly due to its highly stringent contig construction strategy, which is not suitable for metagenome datasets that do not usually have sufficient coverage for most species in the community.

At the time of this manuscript, we became aware of several algorithms for the metagenomic data assembly very recently published. The first one is named Genovo, which deal with the metagenomic data assembly, based on a probabilistic model of read generation (Laserson *et al*., 2011). As Genovo focused on 454 reads data, we compared MAP with Genovo on our 200 bp datasets. However, because of the time-consuming iteration algorithm, Genovo requires too much computing time and uses much more memory than MAP. We have to use a small data of about half size to run Genovo on our machine (20 G memory limit). We constructed three small metagenomic datasets by randomly selecting half of the reads from three 200 bp and error sequencing type datasets, representing HC, MC and LC, respectively. We compared the performance of MAP to Genovo on these three synthetic metagenome datasets. Because the assembled reads are not traceable from the contigs output by Genovo, we cannot calculate the assembled reads and chimeric contigs for Genovo. Thus, to compare the performance of MAP and Genovo, we only use the contig number and contig average length to assess the assembly capability and use the contig sequence base accuracy to assess the assembly accuracy. The results are listed in Supplementary Table S3. MAP generates longer contig size than Genovo does, with a lower contig number than Genovo does. As for contig accuracy, MAP totally outperforms Genovo by producing much less proportion of divergent contig.

The result demonstrates that MAP has also a higher performance than Genovo. Another metagenomic assembly algorithm recently published is Meta-IDBA (Peng *et al*., 2011), which uses de Bruijin graph strategy and targets short reads. The current version did not work on the long reads data (≥200 bp) in our experiments. Thus, we could not compare it with MAP on our test datasets. The last work is a metagenomes scaffolder, named Bambus 2 (Koren *et al*., 2011), which uses mate pairs to merge unitigs into longer contigs and determine the arrangements of contigs along the genomes. Instead of the linear scaffolds, which are generated by the single-genome scaffolder, Bambus 2 outputs graphs that maintain the genomic variation information for metagenomic data. As MAP does not contain a scaffold module, we compared the contigs from MAP to the contigs extracted from the linearized scaffolds generated by finding the longest sequence reconstruction through each scaffold graph by Bambus 2 itself. The three 800 bp and sequencing error datasets are first assembled by the assembler Minimus (Sommer *et al*., 2007) and followed by Bambus 2. The results show that the contigs from MAP have a little lower severe chimeric contig ratio and divergent contig ratio than Bambus 2 on the testing sets (see Supplementary Table S6).

We also tested MAP on a real metagenomic sequencing data composed of a wide diversity of bacteria sampled from the farm soil (available as NCBI Trace Archive Project ID 13699). This typical real data include 100 Mb Sanger shotgun sequencing reads (∼130 000 reads; Tringe *et al*., 2005). The results show that MAP has a similar contig size and contiguity with Celera (see Supplementary Table S7). Because of lack of full information of species references from the real data, the assembling accuracies are not available. However, as we have demonstrated the high accuracy of MAP on the simulated datasets, we believe that MAP can also produce accurate contigs on the real data.

In our test with 20 G memory usage, MAP costs a little more runtime than other assemblers do. For example, on the 800 bp datasets, which have ∼130 k reads on average, MAP costs ∼60 min runtime on average and Celera costs ∼40 min runtime on average. Although for the 200 bp datasets, which have ∼500 k reads on average, MAP costs ∼130 min on average and Celera cost ∼100 min on average.

In summary, with the test of a series of simulated metagenomic datasets, we show that the total assembly performance of MAP can be superior to both Celera and Phrap for typical longer reads by Sanger sequencing and has an evident advantage over Celera, Newbler or the newest Genovo, for typical shorter reads by 454 sequencing.

4 DISCUSSION

Compared with other assemblers, several distinct features of MAP algorithm should be pointed out. First, MAP does not refer to any other information such as genome length or sequencing coverage that is often used in the assemblers targeting the isolated genomes, because such information is clearly not applicable to the situation of metagenomic assembly. What is more important is that MAP employs mate-paired information different from other assemblers did. For example, the Celera Assembler (Myers *et al*., 2000) used mate-paired information in the scaffold constructing. The Celera Assembler later developed a new pipeline CABOG, which finds the best overlap graph in the unitigger module (Miller *et al*., 2008). In this algorithm, mate pairs are used to correct the misassemblies by breaking the unitigs that are found violated with the mate pair constrains. PCAP (Huang *et al*., 2003) used mate-paired information to correct contigs and to link contigs into scaffolds. Different from these assemblers, MAP uses mate pairs as a core measure to construct contigs when repeats hamper the assembly. Based on mate-paired information, MAP designs a series of procedures to implement the layout stage. With respect to the OLC strategy, MAP uses the strategy similar to Celera. Thus in many cases, MAP generates contigs in similar size as Celera does. However, the former uses mate pair information to prevent error contigs from being formed, whereas the latter uses the information to correct and break wrong contigs. These two strategies are quite distinct, but both can avoid errors and in some cases have the similar results. However, the MAP strategy is more reasonable to avoid error when compared with Celera. For example, lacking sufficient mate pair information, Celera was not able to detect the potential errors, which generated through other operation (such as best overlap graph strategy in CABOG) by its method, whereas MAP will sacrifice the size but avoid potential errors by leaving alone the conservative unitigs. In addition, MAP does not include the scaffolder module as Bambus 2 does. Bambus 2 uses the scaffold graph to report the regions of variation in closely related strains and to distinguish between them. The analysis of the strain variation among metagenomes could be carried out by a stand-alone software such as Strainer (Eppley *et al*., 2007).

We have demonstrated that our strategy can achieve higher assembly accuracy in terms of both reads and nucleotide sites, while maintaining high assembly capacity. However, the application of MAP algorithm is sensitive to several factors such as the mate pair length and sequence coverage, which are the state of the art challenges in metagenomic assembly. As the MAP algorithm finds mate pair paths in the overlap graph, too long length of the mate pairs may lead to too many nodes traversed between them and complicating the process of finding mate pair paths. In addition, low sequence coverage may greatly reduce the number of overlaps, making the overlap graph being sparse. Thus, a deeper sequencing may increase the efficiency of MAP, especially in the short read data. It should be noted that without mate pair information, MAP assembles the reads into unitigs, functioning similar to the assembler Minimus (Sommer *et al*., 2007).

It is noted that unlike the metagenome projects sequencing on Sanger platform in early years, recently emerged metagenome projects using 454 sequencing platform were sometimes inclined not to take the mate pair approach to sequence the DNA fragments mainly due to the lower costs. However, robust mate pair protocols are increasingly regarded as essential to new sequencing technologies for their wider application in *de novo* sequencing, especially to metagenomic assembly process (Kunin *et al*., 2008; Miller *et al*., 2010; Pop *et al*., 2007). The study in this article demonstrates that using mate pair information to resolve the repeat in the overlap graph is the promising method for metagenomic assembly. Moreover, our attempt in this article provides valuable support for the mate pair approach in the trade-off in sequencing protocol selection for 454 sequencing platform or other next generate sequencing platform in metagenome projects.

As we have clarified, our assembly method are designed to address the issue of longer metagenomic reads (recommended as >200 bp) from such as Sanger and 454 sequencing technologies. The strongest reason for this interest is that longer reads are

demonstrated more informative to be processed, whereas shorter reads have a significantly reduced ability, although currently the low cost and high throughput of shorter reads sequencing seem to be an attractive option for metagenomic studies (Wommack *et al*., 2008). Moreover, new technologies and sequencing chemistry improvements are expected to further increase high-throughput read lengths, for example, to multi-kilobase length, thus leading to the application in metagenomics studies (Rodrigue *et al*., 2010). It is noted that the runtime of the MAP mainly depends on the times of pairwise overlap calculation, which are determined by the number of reads and the coverage depth. Current short reads sequencing approach uses high-throughput sequencing to obtain much higher coverage depth to compensate the drawback of the short length, which largely increases the times of overlap calculation, thus is not suitable for MAP or other overlap graph assemblers. However, the increase of the read length should meanwhile reduce the required sequencing coverage. Besides, the computer specification keeps improving, and at some point, what may seem extreme today will be within reach in the future. Nonetheless, the improvement to the MAP algorithm is worth doing to reduce the runtime in the future. Thus, we may expect that such a tool as MAP will be well applied to more sequencing platforms.

In view of the complications and challenges presented by metagenome sequence processing, there seems to be no perfect solution to assembling metagenomic datasets. It is, however, non-trivial to further develop novel method to increase assembly quality for complex metagenomes. It is hoped that the algorithm and the resulting tool MAP introduced in this article are shown to meet these challenges.

ACKNOWLEDGEMENTS

We thank Prof. Chunting Zhang of Tianjin University, Prof. Xuegong Zhang of Tsinghua University and for interest to the project and useful discussions. We also thank Dr Xiaobin Zheng, Dr Hong Kang, Yongchu Liu, Jiantao Guo, Luying Liu and Xiaoqi Wang for helpful discussions.

Funding: National Natural Science Foundation of China (30970667, 30770499, 11021463 and 61131003), National Basic Research Program of China (2011CB707500) and Excellent Doctoral Dissertation Supervisor Funding of Beijing (YB 20101000102).

Conflict of Interest: none declared.

REFERENCES

Altschul,S.F. *et al.* (1990) Basic local alignment search tool. *J. Mol. Biol.*, **215**, 403–410.
Chaisson,M. *et al*. (2009) de novo fragment assembly with short mate-paired reads: does the read length matter? *Genome Res*., **19**, 336–346.
Eppley,J.M. *et al*. (2007) Strainer: software for analysis of population variation in community genomic datasets. *BMC Bioinformatics*, **8**, 398.
Gill,S.R. *et al.* (2006) Metagenomic analysis of the human distal gut microbiome. *Science*, **312**, 1355–1359.
Hu,G. *et al*. (2009) MetaTISA: metagenomic translation initiation site annotator for improving gene start prediction. *Bioinformatics*, **25**, 1843–1845.
Huang,X. *et al*. (2003) PCAP: a whole-genome assembly program. *Genome Res.*, **13**, 2164–2170.
Jaffe,D.B. *et al*. (2003) Whole-genome sequence assembly for mammalian genomes: arachne 2. *Genome Res.*, **13**, 91–96.
Kent,J.K. (2002) BLAT—the BLAST-like alignment tool. *Genome Res.*, **12**, 656–664.
Korbel,J.O. *et al.* (2007) Paired-end mapping reveals extensive structural variation in the human genome. *Science*, **318**, 420–426.
Koren,S. *et al*. (2011) Bambus 2: scaffolding metagenomes. *Bioinformatics*, **27**, 2964–2971.
Kunin,V. *et al*. (2008) A bioinformatician's guide to metagenomics. *Microbiol. Mol. Biol. Rev.*, **72**, 557–178.
Laserson,J. *et al*. (2011) Genovo: de novo assembly for metagenomes. *J. Comput. Biol.*, **18**, 429–443.
Li,R. *et al*. (2010) de novo assembly of human genomes with massively parallel short read sequencing. *Genome Res.*, **20**, 265–272.
Li,S. *et al*. (2004) LUCY2: an interactive DNA sequence quality trimming and vector removal tool. *Bioinformatics*, **20**, 2865–2866.
Margulies,M. *et al*. (2005) Genome sequencing in microfabricated high-density picolitre reactors. *Nature*, **437**, 376–380.
Mavromatis,K. *et al.* (2007) Use of simulated datasets to evaluate the fidelity of metagenomic processing methods. *Nat. Methods*, **4**, 495–500.
Metzker,M.L. (2010) Sequencing technologies – the next generation. *Nat. Rev. Genet.*, **11**, 31–46.
Miller,J.R. *et al*. (2008) Aggressive assembly of pyrosequencing reads with mates. *Bioinfomaticts*, **24**, 2818–2824.
Miller,J.R. *et al*. (2010) Assembly algorithms for next-generation sequencing data. *Genomics*, **95**, 315–327.
Mullikin,J.C. *et al*. (2003) The phusion assembler. *Genome Res.*, **13**, 81–90.
Myers,E.W. (2005) The fragment assembly string graph. *Bioinformatics*, **21**, ii79–ii85.
Myers,E.W. *et al.* (2000) A whole-genome assembly of *Drosophila*. *Science*, **287**, 2896–2204.
Peng,Y. *et al*. (2011) Meta-IDBA: a de novo assembler for metagenomic data. *Bioinformatics*, **27**, i94–i101.
Pignatelli,M. *et al*. (2011) Evaluating the fidelity of de novo short read metagenomic assembly using simulated data. *PLoS ONE*, **6**, e19984.
Pop,M. *et al*. (2004) Comparative genome assembly. *Brief. Bioinformatics*, **5**, 237–248.
Pop,M. *et al.* (2007) Bioinformatics challenges of new sequencing technology. *Trends Genet.*, **24**, 133–141.
Pop,M. (2009) Genome assembly reborn: recent computational challenges. *Brief. Bioinformatics*, **10**, 354–366.
Rausch,T. *et al.* (2009) A consistency-based consensus algorithm for de novo and reference-guided sequence assembly of short reads. *Bioinformatics*, **25**, 1118–1124.
Richter,D.C. *et al*. (2008) MetaSim—a sequencing simulator for genomics and metagenomics. *PLoS ONE*, **3**, e3373.
Rodrigue,S. *et al*. (2010) Unlocking short read sequencing for metagenomics. *PLoS ONE*, **5**, e11840.
Schatz,M.C. *et al*. (2010) Assembly of large genomes using second-generation sequencing. *Genome Res.*, **20**, 1165–1173.
Sommer,D.D. *et al.* (2007) Minimus: a fast, lightweight genome assembler. *BMC Bioinformatics*, **8**, 64.
Tasse,L. *et al.* (2010) Functional metagenomics to mine the human gut microbiome for dietary fiber catabolic enzymes. *Genome Res.*, **20**, 1605–1612.
Tringe,S.G. *et al*. (2005) Comparative metagenomics of microbial communities. *Science*, **308**, 554–557.
Tyson,G.W. *et al.* (2004) Genomic structure and metabolism through reconstruction of microbial genomes from the environment. *Nature*, **428**, 37–43.
Venter,J.C. *et al*. (2004) Environmental genome shotgun sequencing of Sargasso sea. *Science*, **304**, 66–74.
Warren,R.L. *et al*. (2007) Assembling millions of short DNA sequences using SSAKE. *Bioinformatics*, **23**, 500–501.
Wommack,K.E. *et al*. (2008) Metagenomics: read length matters. *Appl. Environ. Microb.*, **74**, 1453–1463.
Zerbinor,D.R. *et al*. (2008) Velvet: algorithms for de novo short read assembly using de Bruijn graphs. *Genome Res.*, **18**, 821–829.
Zhu,W. *et al*. (2010) *Ab initio* gene identification in metagenomic sequences. *Nucleic Acids Res.*, **38**, e132.