PeerJ

An extended genovo metagenomic assembler by incorporating paired-end information

Afiahayati, Kengo Sato and Yasubumi Sakakibara

Department of Biosciences and Informatics, Keio University, Hiyoshi, Kohoku-ku, Yokohama, Japan

ABSTRACT

Metagenomes present assembly challenges, when assembling multiple genomes from mixed reads of multiple species. An assembler for single genomes can't adapt well when applied in this case. A metagenomic assembler, Genovo, is a de novo assembler for metagenomes under a generative probabilistic model. Genovo assembles all reads without discarding any reads in a preprocessing step, and is therefore able to extract more information from metagenomic data and, in principle, generate better assembly results. Paired end sequencing is currently widely-used yet Genovo was designed for 454 single end reads. In this research, we attempted to extend Genovo by incorporating paired-end information, named Xgenovo, so that it generates higher quality assemblies with paired end reads.

First, we extended Genovo by adding a bonus parameter in the Chinese Restaurant Process used to get prior accounts for the unknown number of genomes in the sample. This bonus parameter intends for a pair of reads to be in the same contig and as an effort to solve chimera contig case. Second, we modified the sampling process of the location of a read in a contig. We used relative distance for the number of trials in the symmetric geometric distribution instead of using distance between the offset and the center of contig used in Genovo. Using this relative distance, a read sampled in the appropriate location has higher probability. Therefore a read will be mapped in the correct location.

Results of extensive experiments on simulated metagenomic datasets from simple to complex with species coverage setting following uniform and lognormal distribution showed that Xgenovo can be superior to the original Genovo and the recently proposed metagenome assembler for 454 reads, MAP. Xgenovo successfully generated longer N50 than Genovo and MAP while maintaining the assembly quality even for very complex metagenomic datasets consisting of 115 species. Xgenovo also demonstrated the potential to decrease the computational cost. This means that our strategy worked well. The software and all simulated datasets are publicly available online at http://xgenovo.dna.bio.keio.ac.jp.

Subjects Bioinformatics, Computational Biology, Genomics
Keywords Genovo, 454 paired end reads, de novo metagenomic assembler

Submitted 9 September 2013
Accepted 10 October 2013
Published 31 October 2013

Corresponding author
Yasubumi Sakakibara,
yasu@bio.keio.ac.jp

Academic editor
Kenta Nakai

Additional Information and Declarations can be found on page 21

DOI **10.7717/peerj.196**

INTRODUCTION

Next generation sequencing (NGS) technologies have allowed an explosion in sequencing with the increased throughput and decrease in cost of sequencing (*Scholz, Lo & Chain,*

2012). The field of metagenomics has adapted to the new type of sequencing technologies which allows us to generate reads from multiple genomes effectively (*Peng et al., 2011*). Although a number of metagenomes have been sequenced using NGS, few studies have reported their assembly results (*Hiatt et al., 2010*; *Namiki et al., 2012*; *Qin et al., 2010*). Metagenomes have presented a number of additional assembly challenges, how to assemble multiple genomes from mixed reads of multiple species. In metagenomic data, the number of genomes and the coverage of each genome are initially unknown. The data potentially consists of multiple genomes with inhomogenous coverage distribution (*Chen & Pachter, 2005*; *Lai et al., 2012*; *Laserson, Jojic & Koller, 2011*; *Nagarajan & Pop, 2013*; *Namiki et al., 2012*; *Peng et al., 2011*; *Scholz, Lo & Chain, 2012*). Assemblers for single genomes can't adapt well when applied in this case (*Lai et al., 2012*; *Laserson, Jojic & Koller, 2011*; *Namiki et al., 2012*; *Peng et al., 2011*; *Scholz, Lo & Chain, 2012*). This assembler generates high rate of misassembled contigs called chimera contig which consists of reads from different species in metagenome assembly (*Lai et al., 2012*; *Mavromatis et al., 2007*; *Pigmatelli & Moya, 2011*).

There are a number of effective assemblers for single genome, but only five attempt to solve metagenome cases: MetaVelvet (*Namiki et al., 2012*), Meta-IDBA (*Peng et al., 2011*), IDBA-UD (*Peng et al., 2012*), MAP (*Lai et al., 2012*) and Genovo (*Laserson, Jojic & Koller, 2011*). Metavelvet, Meta-IDBA and IDBA-UD use the De Bruijn graph approach. They were designed to handle short read data. IDBA-UD is an extension of Meta-IDBA solving uneven sequencing depths of different regions of genomes from different species (*Peng et al., 2012*). MAP was designed for longer reads produced by Sanger (700–1000 bp) and 454 sequencing technology (200–500 bp). MAP uses an improved OLC (Overlap/Layout/Consensus) strategy integrating mate pair information (*Lai et al., 2012*). Genovo was designed for longer reads of 454 sequencing data; it is a metagenomic assembler under a generative probabilistic model (*Laserson, Jojic & Koller, 2011*).

Unlike other methods, Genovo assembles all reads without discarding any reads. It doesn't detect and correct read errors in a preprocessing step. This avoids filtering out any low coverage genomes, hence hopefully is able to extract more information from metagenomic data in order to generate better assembly results. The consequence is high computational cost (*Laserson, Jojic & Koller, 2011*). Paired end sequencing is currently widely-used yet Genovo was designed for single end reads. In this research, we extend Genovo by incorporating paired-end information, named Xgenovo. We also design algorithms to decrease the computational cost. We modified some procedures of Genovo in determining the location of a read in the coordinate system of contig and offset (the beginning of the read) so that it generates higher quality assemblies with paired end reads. Genovo uses Chinese Restaurant Processes (CRP) to get prior accounts of the unknown number of genomes in the sample. First, we modified CRP by adding a bonus parameter which intends for a pair of reads to be in the same contig also as an effort to solve chimera contig case. Second, we used relative distance for the number of trials in the symmetric geometric distribution instead of using distance between the offset and the center of the contig used in Genovo. For paired end reads, this process should take into account the

insert length parameter. Using this relative distance, a read sampled in the appropriate location has higher probability. Therefore a read will be mapped in the correct location.

We used Metasim (*Richter et al., 2008*) to generate simulated metagenomic datasets. In order to measure the performances more comprehensively, we applied two kinds of species coverage (abundance) distribution for the dataset, uniform and log-normal distribution. In total, we generated 16 simulated datasets from simple to complex datasets. We compared the performance of Xgenovo with the naive use of the original Genovo and the recently proposed matagenome assembler for 454 reads, MAP, which also utilizes paired end information. MAP outperforms standard single genome assemblers for 454 reads, Celera (*Myers et al., 2000*; *Miller et al., 2008*) and Newbler (*Margulies et al., 2005*). In this research, Xgenovo was not compared with single genome and metagenome assemblers which are designed for Illumina types of short read data (<100 bp), like Velvet (*Zerbino & Birney, 2008*), SOAPdenovo (*Li et al., 2010*), IDBA (*Peng et al., 2010*), MetaVelvet (*Namiki et al., 2012*), Meta-IDBA (*Peng et al., 2011*) and IDBA-UD (*Peng et al., 2012*). Xgenovo generated longer N50 than the original Genovo and MAP while maintaining the assembly quality for all datasets. Xgenovo also demonstrated the potential to decrease the computational cost. We successfully extended Genovo by incorporating paired-end information so that it generates higher quality assemblies with paired end reads by modifying Genovo in determining the location of a read in the coordinate system of contig and offset (the beginning of read), different from other assemblers (*Koren, Treangen & Pop, 2011*; *Li et al., 2010*; *Namiki et al., 2012*; *Peng et al., 2012*; *Zerbino & Birney, 2008*; *Zerbino et al., 2009*) which used paired end information to generate scaffolds. The software and all simulated datasets are publicly available online at http://xgenovo.dna.bio.keio.ac.jp.

METHODS

Overview of Genovo

Genovo is a metagenomic assembler under a generative probabilistic model, illustrated in Fig. 1. An assembly is represented as a list of contigs and a mapping of each read to the contigs. Each contig is represented as a list of DNA letters $\{b_{so}\}$, where b_{so} is the letter at position o of contig s. The mapping represents the position of each read x_i in a coordinate system of contigs and offsets, each read has its contig number s_i and its offset o_i (starting location of the read within the contig). Each read mapped to the contig is aligned base-for-base denoted by y_i . To represent a set of variables, bold-face letters is used, for example, $\boldsymbol{b}$ represents a set of DNA letters. The probabilistic model is described as below:

1. Construct infinite number of contigs consisting of infinitely many DNA letters. Assume that there are infinitely many contigs consisting of an infinite number of DNA letters sampled following uniform distribution, shown in Fig. 1. Because of the finite number of reads, only a finite number of infinitely many contigs will have reads mapped to them.
2. Map each read to the contigs.

There is a coordinate system of contigs and offsets showing the position of reads mapped to the contigs. Two steps are used in mapping process: first, partition the reads (N) to clusters

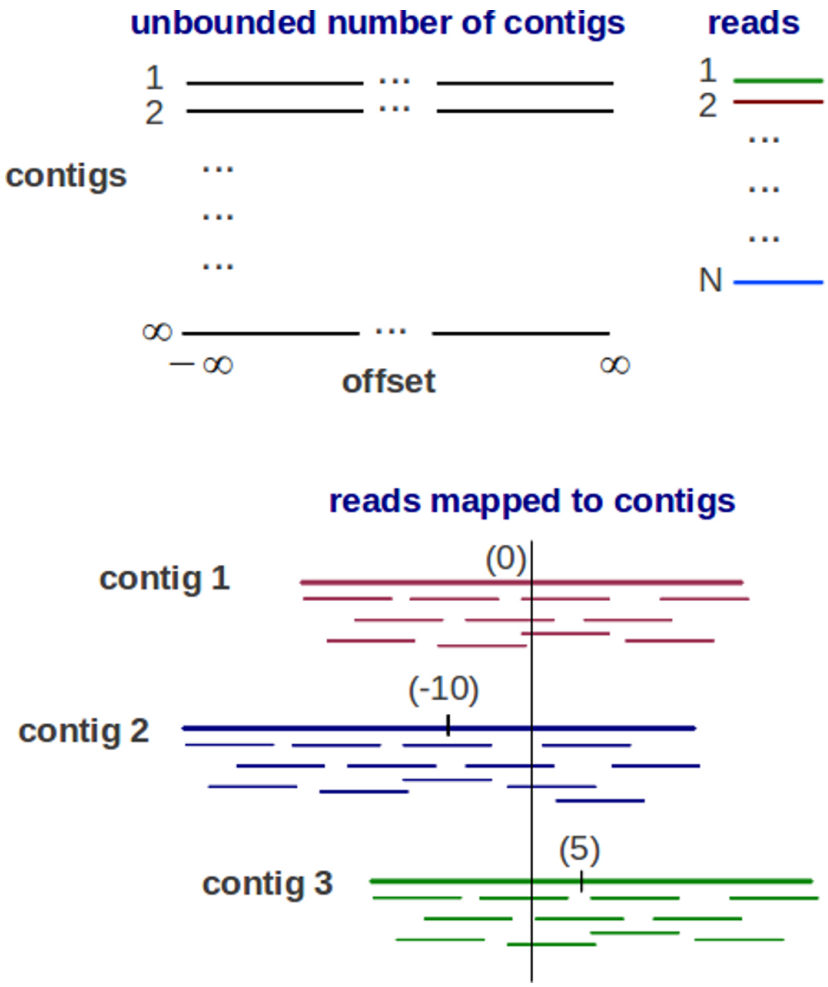

Figure 1 The Generative probabilistic model in Genovo. There are an unbounded number of contigs constructed with unbounded length (from negative infinity to positive infinity) and N reads. The reads are mapped to 3 contigs in a coordinate system of contigs and offsets (the beginning location of a read). Each contig has its center. The center of contig 1 is 0, contig 2 is −10 and contig 3 is 5.

using CRP shown in (1). The number of clusters represents the number of contigs (s) as an initial number of multiple genomes. The parameter α controls the expected number of classes.

$$s \sim CRP(\alpha, N) \tag{1}$$

Second, assign each cluster of reads to a contig. A good contig is defined as a contig having the most reads towards the center of the contig. Therefore, a starting point of read o_i within each contig is assigned using a symmetric geometric distribution, shown in (2). The parameter ρ_s controls the length of a contig.

$$o_i \sim G(\rho_s) \quad \forall i = 1..N \tag{2}$$

3. Copy the letters of each read x_i (with some noises) to the mapped location in contigs starting from position o_i with orientation, insertion and deletion encoded by alignment $y_{i,}$ shown in (3), l_i is the length of $read_i$, ρ_{ins} is the probability of insertion, ρ_{del} is the probability of deletion, ρ_{mis} is the probability of incorrect copying (mismatch) and A is the distribution representing the noise model known for the sequencing technology.

$$x_i, y_i \sim A(l_i, s_i, o_i, b, \rho_{ins}, \rho_{del}, \rho_{mis}) \tag{3}$$

To generate appropriate assemblies, Genovo performs a series of iterated hill climbing procedures, maximizing or sampling local conditional probabilities to reach MAP solution (the best likelihood), illustrated in Fig. 2. This algorithm is run until convergence (200–300 iterations). Genovo outputs the best assembly, the model with the highest probability during the iterations. The likelihood of this model consists of the likelihood of the alignments $\log p(\boldsymbol{x},\boldsymbol{y}|\boldsymbol{s},\boldsymbol{o},\boldsymbol{b})$, the likelihood for generating (uniformly) each contig letter $\log p(\boldsymbol{b})$, the likelihood of contigs $\log p(\boldsymbol{s})$, and the likelihood of offsets $\log p(\boldsymbol{o}|\boldsymbol{s},\rho)$, shown in (4)– (8), where S is the number of contigs, N_s is the number of reads in contig s, L is the total length of all contigs, ρ_s is the control parameter of the length of a contig, β is the count of DNA character = 4, $score^i_{READ}$ is the alignment score of $read_i$ mapped to the contig, and

$$O_s = \sum_{k=1}^{N_s} |o_k|$$

$$\log p(x,y|s,o,b) + \log p(b) + \log p(s) + \log p(o|s,\rho) \tag{4}$$

$$\log p(x,y|s,o,b) = \sum score^i_{READ} \tag{5}$$

$$\log p(b) = -\log|\beta|L \tag{6}$$

$$\log p(s) = S\log(\alpha) + \sum_{i=1}^{S} \log\Gamma(N_s) + const(\alpha, N) \tag{7}$$

$$\log p(o|s,\rho_s) = \sum_{i=1}^{S} \left[O_s \log(1-\rho_s) + N_s \log\rho_s + const(N)\right] \tag{8}$$

The procedures are described as below:

1. Consensus contig

   This procedure attempts to increase the likelihood of alignment by updating over the DNA letter variable in contigs b_{so}. The letters of prior contigs are sampled following uniform distribution, therefore the likelihood is maximized by tuning up the number of reads in the current assembly which align the letter $b \in B$ to the location (s,o). $b_{so} = argmax_{b\in B} a^b_{so}$ where a^b_{so}.
2. Read mapping

   This procedure is the main procedure in Genovo: moving reads to the more appropriate location. It performs stochastic ICM updates over the read variables s_i, o_i,

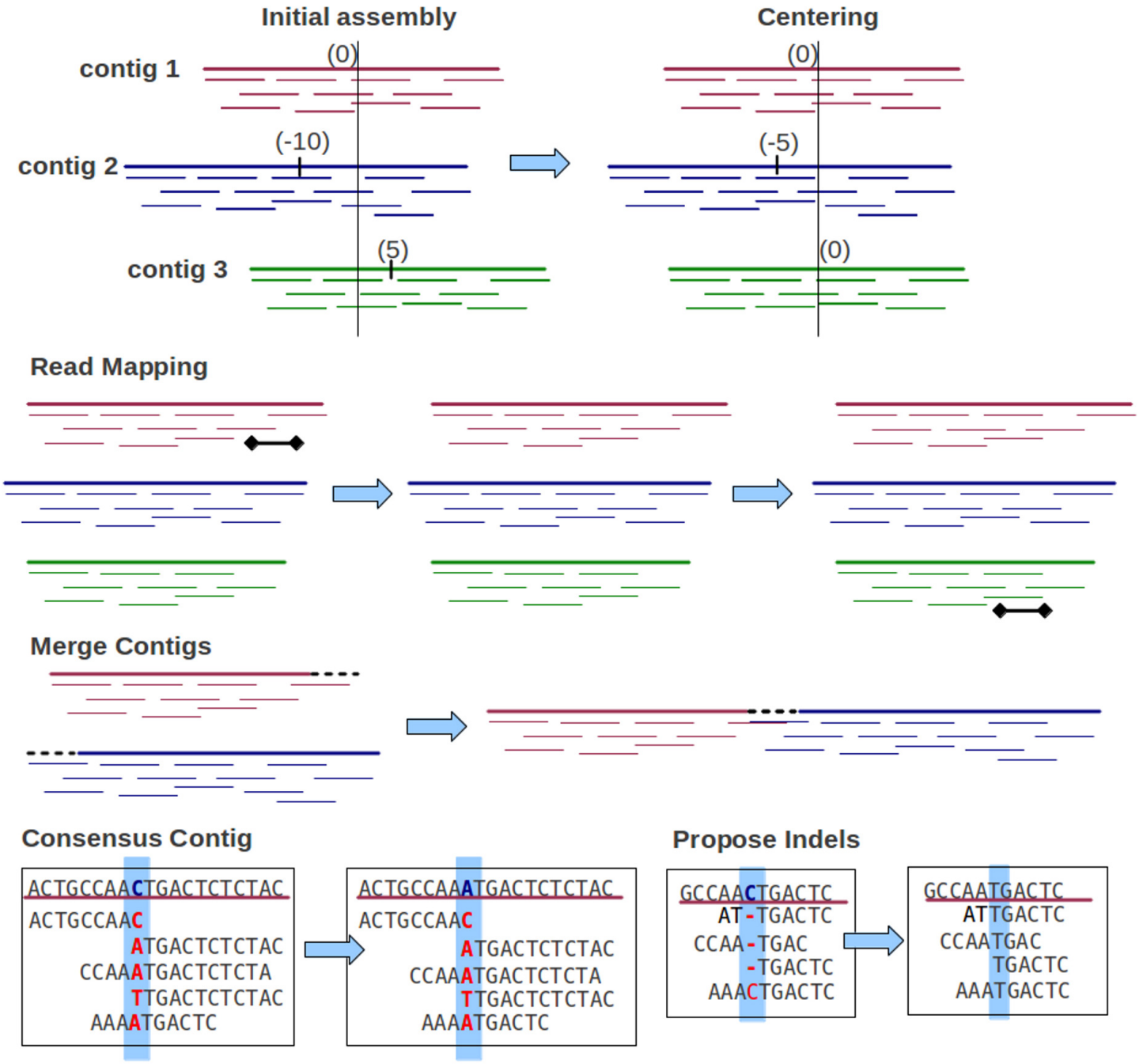

Figure 2 Iterative procedures of Genovo. 5 iterative procedures are illustrated. The centering procedure makes the center of contigs towards zero: 3 contigs with centers (0, −10, 5) become (0, −5, 0) after centering. The read mapping procedure moves reads to the more appropriate location: a read in contig 1 is moved to contig 3. The merge procedure merges two contigs: contig 1 and contig 2 are merged. The consensus contig procedure updates the letter in the consensus contig: letter C in the consensus is changed to A. The propose indels procedure: most reads have an insertion, therefore it is proposed to delete the corresponding letter (C) in the contig and realign the reads.

y_i sequentially for each read x_i. First, read x_i is removed, then a new location is sampled from the joint posterior $p(s_i = s, o_i = o, y_i = y | x_i, \boldsymbol{y}_{-i}, \boldsymbol{s}_{-i}, \boldsymbol{o}_{-i}, \boldsymbol{b}, \rho)$.

3. Global moving

   These procedures change a set of variables at once which speed up convergence. These procedures consist of:

   (a) Propose indels

   If most mapped reads have an insertion at a specific location then the deletion of the corresponding letter in the contig will be proposed and the reads will be realigned. While vice versa, if most mapped reads have a deletion at a specific location, the insertion will be proposed. If the likelihood improves, the proposal will be accepted.

   (b) Centering the contigs

   Each contig has a center. A good contig is defined as a contig having the center towards zero. This procedure shifts the coordinate system of each contig to maximize the likelihood of offset by making the center of the contigs towards zero. In the illustration shown in Fig. 2, there are 3 contigs. After implementing this procedure, the center of each contig shifts towards zero.

   (c) Merge

   It is common for two contigs to have overlapping ends. The assembly created when merging two such contigs would have a higher probability of the model, but if the assembly is only generated by the "read mapping" procedure, it requires multiple iterations. If the end of a contig overlaps with the beginning of another contig, then Genovo will align those ends, the reads in the overlapping area are re-aligned and both contigs are merged. This procedure will be executed if it improves the likelihood of model.

   (d) Chimeric read solving

   Chimeric reads are reads having two segments of length >20 that mapped to noncontiguous areas of the reference genome (*Lasken & Stockwell, 2007*). The Genovo algorithm assumes that these reads often reach the end of an assembled contig. To solve this case, Genovo disassembles the reads assembled in the end of a contig occasionally (every 5 iterations). Using this procedure, other correct reads or contigs can merge with it and the likelihood of model will increase. If a disassembled read is not chimeric, it will be reassembled appropriately in the next iteration and the likelihood of model will be maintained like the previous iteration.

Extended Genovo

We extended Genovo by modifying some procedures in order to fit in with paired end reads incorporating paired-end information; this model is called Xgenovo. First, we modified CRP by adding a bonus parameter. Second, we modified the sampling process of the location of a read in a contig. Xgenovo doesn't use the chimeric read solving procedure from Genovo because it will decrease the likelihood of model. In the extended model, greater numbers of pairs of reads in the contigs increase the likelihood of model. In the

chimeric read solving procedure, the reads assembled in the end of a contig disassemble occasionally. The reads may be mates to other reads in a contig, the number of pairs of reads will decrease therefore the likelihood of model also will decrease.

Modified CRP

Genovo uses CRP to cluster the reads. The concept of CRP is that the rich get richer. The probability of the new customer sitting at an occupied table is proportional to the number of customers already sitting at it and the probability of the new customer sitting at the next unoccupied table is proportional to a concentration parameter, α, represented by (9). In the assembly case, a customer is a read while a table is a contig. The concentration parameter determines the intention of a new customer sitting at a new table. The customer inclines to sit at the most popular tables (*Johnson, 2012*). A CRP is a conditional distribution which is invariant to the order of the items (*Aldous, 1985*) which, in our case, are the reads.

$$p(s_i = s|s_{-i}) \sim \begin{cases} N_{-i,s} & s\text{: an existing contig} \\ \alpha & s\text{: new contig} \end{cases} \tag{9}$$

$N_{-i,s}$ counts the number of items, not including i, that is in contig s. For paired end reads, beside being concerned with the concept that the rich get richer, it should also care whether a pair of reads are in the same contig. Therefore, we give a bonus if a read is in the same contig with its mate. In the illustration shown in Fig. 3A, a read chooses a contig in single read case. There are 3 contigs (Contig I, Contig II and Contig III) with reads (4, 2, 2) and a new contig (Contig IV) can be created. The contig which will be chosen depends on the number of reads in the contig and the concentration parameter, α, so that the candidate contigs are contig I (having the most read) and contig IV.

In the paired end read case, illustrated in Fig. 3B, aside from the number of reads in the contig and the concentration parameter, it should also depend on the bonus parameter, represented by (10). This bonus parameter (β) intends for a pair of reads to be in the same contig and as an effort to solve chimera contig case. Therefore the candidate contigs are contig I (having the most reads), contig III (having its mate) and contig IV. If the 2nd read is mapped in Contig III, a bonus will be given.

$$p(s_i = s|s_{-i}) \sim \begin{cases} N_{-i,s} * \beta & s\text{: a mate contig} \\ N_{-i,s} & s\text{: an existing contig} \\ \alpha & s\text{: new contig} \end{cases} \tag{10}$$

Modified sampling process of an offset

Sampling process of an offset means assigning a location of the read's offset at a contig. Geometric distribution represents the probability distribution of the number $y = x - 1$ of failures before the first success, shown in (11), p is the probability on each trial and k is the number of trials (*Degroot & Schervish, 2011*).

$$P(x = k) = (1 - p)^k p \tag{11}$$

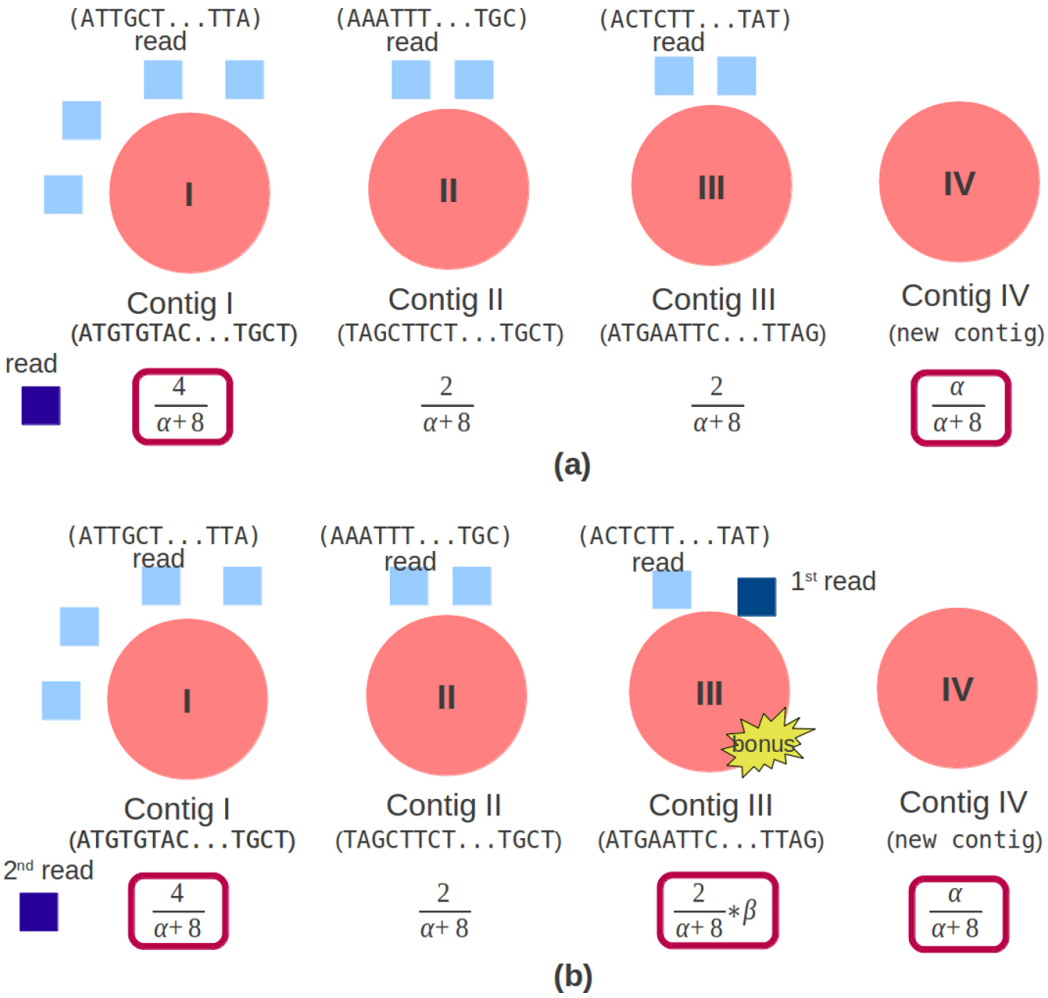

Figure 3 Original and modified CRP. The red circle is a contig, the light blue square is a read mapped in the contig. (A) Original CRP: the dark blue square is a new read which will be mapped. There are 3 contigs (Contig I, Contig II and Contig III) with reads (4, 2, 2) and 1 new contig (Contig IV). (B) Modified CRP: the dark blue square is paired end reads, the 1st read is mapped in Contig III and the 2nd read will be mapped. There are 3 contigs (Contig I, Contig II and Contig III) with reads (4, 2, 2) and 1 new contig (Contig IV). If the 2nd read is mapped in Contig III, a bonus will be given.

Genovo uses this concept. Sampling the beginning of a read (an offset) at a location x means that Genovo get failures for sampling an offset at location 1 until $x-1$ and success at location x. Genovo uses the negative and positive integer for the offsets representation in the contigs. A good contig is defined as a contig having the most reads towards the center of contigs. Therefore Genovo uses a symmetric variation of geometric distribution that includes all the negative integers with a center at 0 to sample a starting point o_i of read within each contig, shown in (12).

$$G(o;\rho_s) = \begin{cases} 0.5(1-\rho_s)^{|o_t|}\rho_s o \neq 0 \\ \rho_s o = 0 \end{cases} \tag{12}$$

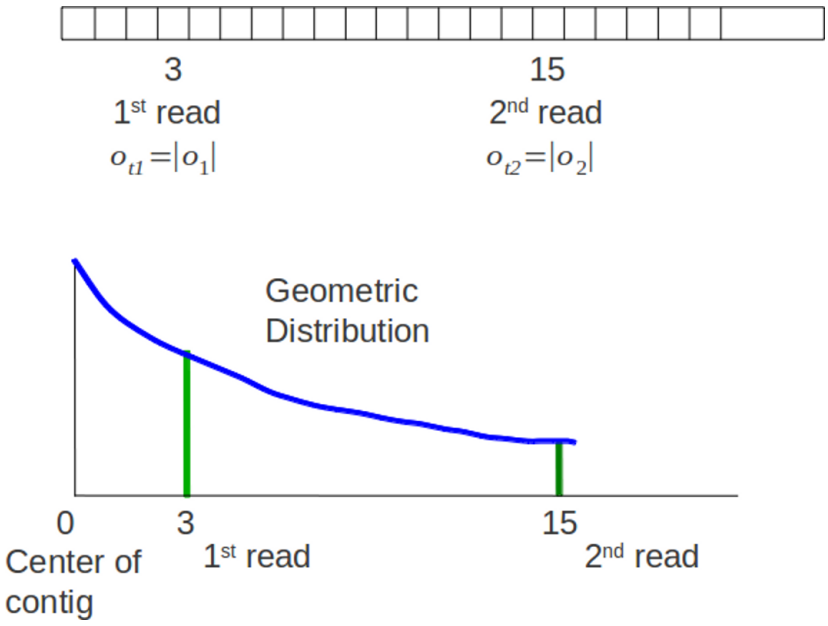

Figure 4 Original sampling process. The 1st read and 2nd read are mapped in a contig. There is a graph of the symmetric geometric distribution of the reads in the positive integer side. The center of the distribution is the center of contig (0).

$(1+N_s, 1+\beta+O_s) = \frac{N_s}{N_s+\beta+O_s}$. The number of trials, $|o_t|$, is the distance between the offset and the center (the absolute value of the offset). The parameter ρ_s controls the length of a contig. This parameter is the same with the probability of success on each trial p in the original geometric distribution. As the posterior distribution of p can be determined if a Beta(α, β) prior is given (*Degroot & Schervish, 2011*), Genovo also uses a known beta distribution to update the value of ρ_s. Genovo sets ρ_s to the mode of the Beta distribution where $O_s = \sum_{k=1}^{N_s} |O_k|$.

For paired end reads, the sampling process of offset should take into account the insert length parameter. Xgenovo uses the relative distance of a read to its mate incorporating the insert length. In the illustration shown in Fig. 4, there are paired end reads with insert length distribution $(\mu, \delta) = (14, 3)$. The 1st read is mapped in the offset 3 and the 2nd read is mapped in the offset 15. Genovo uses the absolute value of the offset as the number of trials, hence the number of trials for the 1st read is 3 and for the 2nd read is 15. From the illustration, we can see that the 2nd read sampled in the appropriate location has lower probability than in the location which is close to the center of the contig. It happens because the center of the symmetric geometric distribution for the 2nd read is the center of the contig and doesn't take into consideration the insert length parameter. While in Xgenovo, the number of trials for the 1st read is the same as Genovo (3) yet relative distance is used for the 2nd read. The relative distance is defined by $|o_1 + \mu - o_2|$. Xgenovo utilizes the insert length parameter to determine the center of the distribution of the 2nd read. Therefore the 2nd read sampled in the appropriate location has higher probability. In the illustration shown in Fig. 5, the number of trials is $o_{t2} = |3 + 14 - 15| = 2$. The formula

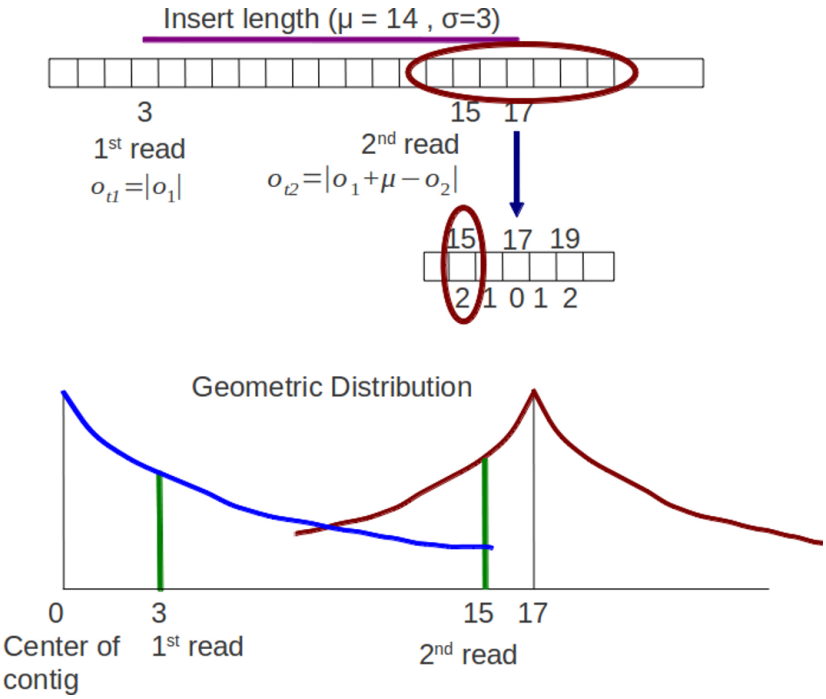

Figure 5 Modified sampling process. The 1st read and 2nd read are mapped in a contig. Insert length distribution is $(\mu, \delta) = (14, 3)$. The distribution of the 1st read and 2nd read have different centers. The center of the 2nd read incorporates the insert length parameter, the center is 3 (the offset of 1st read) + 14 (insert length) = 17.

of symmetric geometric distribution for the 1st read is same with Genovo shown in (13), while the distribution for the 2nd read is shown in (14).

$$G(o_1|\rho_{1s}) = \begin{cases} 0.5(1-\rho_{1s})^{|o_1|}\rho_{1s} & o_1 \neq 0 \\ \rho_{1s} & o_1 = 0 \end{cases} \tag{13}$$

$$G(o_{2s}|o_1, o_2, \rho_{2s}) = \begin{cases} 0.5(1-\rho_{2s})^{|o_{t2}|}\rho_{2s} & o_{t2} \neq 0 \\ \rho_{2s} & o_{t2} = 0 \end{cases} \quad \text{where } o_{t2} = |o_1 + \mu - o_2| \tag{14}$$

There is a possibility that the 2nd read is not sampled in the same contig with the 1st read. For this case, both the 1st read and the 2nd read are considered as 1st read (single read). There are two ρ_s, ρ_{1s} for the 1st read and ρ_{2s} for the 2nd read. Both are updated using known Beta $(1 + N_{1s}, 1 + \beta + O_{1s}) = \frac{N_{1s}}{N_{1s}+\beta+O_{1s}}$ distributions. The ρ_{1s} is updated by the mode of distribution Beta where $O_{1s} = \sum_{k=1}^{N_{1s}} |o_{1k}|$. The ρ_{2s} is updated by the mode of $(1 + N_{2s}, 1 + \beta + O_{t2s}) = \frac{N_{2s}}{N_{2s}+\beta+O_{t2s}} O_{t2s} = \sum_{k=1}^{N_{2s}} |o_{t2k}|$ distribution Beta where N_{1s} is the number of the 1st read or single read (read which is not in the same contig with its mate) in contig s, o_1 is the offset of a read, N_{2s} is the number of the 2nd read in contig s and o_{t2} is the number of trial for 2nd read. By using this relative distance, reads sampled in the appropriate location in a contig has a higher probability of model so that a contig produced is correct compared to using default distance in Genovo.

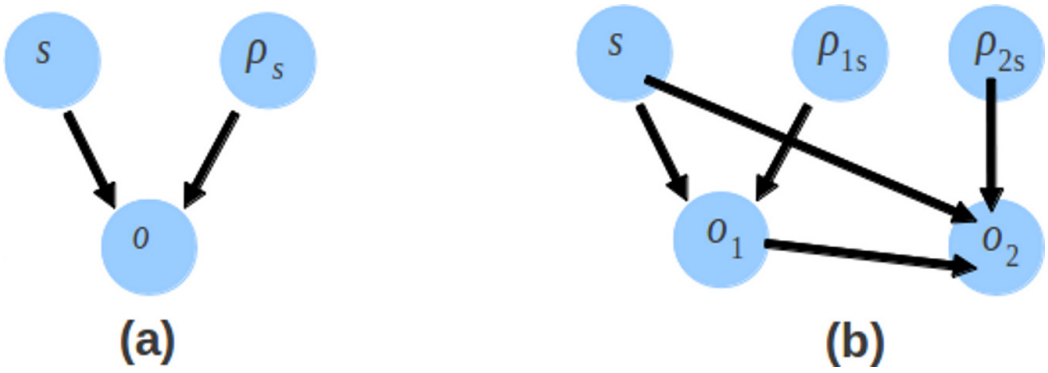

Figure 6 The directed graphical model representing the likelihood of offsets. The likelihood of offsets for the original Genovo (A) in the left side with 3 nodes, and the extended Genovo (B) in the right side with 5 nodes.

Likelihood

The probability distribution in CRP and sampling process are changed so that the likelihood of the model also changes. Like Genovo, the likelihood of our model also consists of 4 components, shown in (4). The likelihood of the alignments $\log p(\boldsymbol{x},\boldsymbol{y}|\boldsymbol{s},\boldsymbol{o},\boldsymbol{b})$ and the likelihood for generating (uniformly) each contig letter $\log p(\boldsymbol{b})$ are the same as Genovo's. While the differences are for the likelihood of contigs, shown in (15) and the likelihood of offsets, shown in (16)–(18).

$$\log p(s) = S\log(\alpha) + \sum_{i=1}^{S} \log\Gamma(N_s) + \log\Gamma(\alpha) - \log\Gamma(N+\alpha) + N_{2s}\log(\beta) \tag{15}$$

$$\log p(o|s,\rho_{1s},\rho_{2s}) = \log p(o_1|s,\rho_{1s}) + \log p(o_2|s,\rho_{2s}) \tag{16}$$

$$\log p(o_1|s,\rho_{1s}) = \sum_{i=1}^{S} [O_{1s}\log(1-\rho_{1s}) + N_{1s}\log\rho_{1s} + N_{1s}\log 0.5] \tag{17}$$

$$\log p(o_2|s,o_1,\rho_{2s}) = \sum_{i=1}^{S} [O_{t2s}\log(1-\rho_{2s}) + N_{2s}\log\rho_{2s} + N_{2s}\log 0.5] \tag{18}$$

where S is the number of contigs, N_s is the number of read in contig s, $O_{1s} = \sum_{k=1}^{N_{1s}} |o_{1k}| O_{t2s} = \sum_{k=1}^{N_{2s}} |o_{t2k}|, N_{1s}$ is the number of 1st read or single end read in contig s and N_{2s} is the number of 2nd read in contig s. There is an additional component for the likelihood of contigs which takes into account the bonus parameter and the number of 2nd reads. The likelihood of offset consists of the likelihood of the offset of the 1st read and the likelihood of the offset of the 2nd read. The directed graphical model representing the likelihood of offsets is shown in Fig. 6. Genovo has 3 variables (o, ρ_s and s), the probability of offset (o) given ρ_s and s. Xgenovo has 5 variables (o_1, o_2, ρ_{1s}, ρ_{2s} and s), the probability of the offset of the 1st read (o_1) given ρ_{1s} and s, the probability of the offset of the 2nd read (o_2) given o_1, ρ_{2s} and s. Like Genovo, Xgenovo also outputs the assembly that achieved the highest likelihood thus far.

RESULTS AND DISCUSSION

We used Metasim (*Richter et al., 2008*) to generate simulated metagenomic datasets. The read length was set at 250 bp and used the default 454 sequencing noise provided by Metasim. The insert length distribution (μ, δ) is (3000, 200). We generated 50,000 pairs of reads for each dataset which is twice the size of the simulated dataset used in Genovo's paper. To evaluate the performances of metagenomic assemblers comprehensively, we applied two kinds of species coverage (abundance) distribution for the dataset, uniform distribution and log-normal distribution. Uniform distribution means that each species in the dataset has the same probability to exist, or it can be said that each species has same abundance value or similar to each other. Second, we applied species abundance following log-normal distribution. The log-normal distribution appropriately describes the microbial abundance distributions (*Unterseher et al., 2011*). We generated simulated metagenomic datasets from simple to complex datasets. The complexity of dataset is based on the number of genomes in the dataset (*Mende et al., 2012*). For log-normal distribution, first we generated the simplest dataset consisting of 13 viruses which is the same complexity with a simulated dataset used in Genovo's paper, with the lowest coverage = 7.42x, the highest coverage = 188.93x, as LC and HC respectively, then the 2nd dataset consists of 17 viruses (LC = 10.82x, HC = 363.18x), the 3rd dataset consists of 30 viruses (LC = 6.64x, HC = 708.79x) and the 4th dataset consists of 35 viruses (LC = 10.59x, HC = 492.23x). For uniform distribution, we generated 4 simulated metagenomic datasets which contain 35 viruses with the same coverage for each species. In the 1st dataset each species has 30 times coverage of the genome sequences; in the 2nd dataset each species has 40; in the 3rd dataset each species has 50 and in the 4th dataset each species has 60.

We compared the performance of Xgenovo with the naive use of the original Genovo and MAP. In the MAP's paper, they used datasets consisting of 113 species therefore, to compare the performance between Xgenovo and MAP more rigorously, we generated very complex datasets consisting of 50 viruses, 60 viruses, 90 viruses and 115 viruses, both for log-normal distribution, 50 viruses (LC = 9.10x, HC = 427.04x), 60 viruses (LC = 3.95x, HC = 648.49x), 90 viruses (LC = 8.46x, HC = 831.79x), 115 viruses (LC = 10.52x, HC = 1986.55x) and for uniform distribution, with the same coverage: 50 viruses (50x), 60 viruses (50x), 90 viruses (40x) and 115 viruses (55x). In total, we generated 16 simulated datasets. The complete descriptions of all datasets are provided in http://xgenovo.dna.bio.keio.ac.jp.

In order to evaluate the assembly capacity, we used four measurements: N50, total length of contig, maximum length of contig and the number of contigs. To evaluate the assembly quality we used two measurements: cover rate and chimera rate. We were also concerned with the computational cost, CPU time and required memory. N50 is a standard statistical measure evaluating the assembly performance which indicates the largest value y such that at least 50% of the genome is covered by contigs of length of $\geq y$. We follow *Namiki et al. (2012)* to measure the cover rate and chimera rate. The cover rate of genome X is defined as the ratio of the total length of contigs which are best aligned to genome X

divided by the length of genome X, shown in (19), where C_i is the length of contig i which is best aligned to genome A.

$$\text{Cover rate of } A = \frac{(\sum |C_i|)}{|A|} \tag{19}$$

To determine whether a contig is chimeric or not: first, the best hit alignments between a contig and the set of input reference genomes using BLAST is calculated; second, if a contig has more than two subsequences that are aligned to different genomes, and those subsequences are longer than 1% of the contig length, the contig is determined to be chimeric.

We compared the performance of Xgenovo with the naive use of the original Genovo and MAP. Genovo set the parameter $\alpha = 2^{35}$, the best parameter value to assemble. To know the performance of Xgenovo, we used combinations of parameters between α and β (bonus). The combinations were $\alpha = 2^{35}$ and $\beta = 0.1\alpha, 0.3\alpha, 0.5\alpha$. We ran both Xgenovo and Genovo for 200 iterations which reaches convergence. We used the default setting for MAP. All computations were executed with Intel(R) Xeon(R) E5540 processors (2.53 GHz) and 48 GB physical memory.

Experiments on different numbers of species with coverage following log-normal distribution

The results for experiments on different numbers of species with coverage following log-normal distribution were shown in Table 1. The results were the best performances of parameter combinations between α and β. Xgenovo generated the highest N50 for all datasets, shown in Fig. 7. Compared to the original Genovo, Xgenovo increased N50 by 28.1% (2473 bp) for the dataset with 13 viruses, increased N50 by 20.3% (7202 bp) for the dataset with 17 viruses, increased N50 by 119.5% (19213 bp) for the dataset with 30 viruses and increased N50 by 75.0% (9112 bp) for the dataset with 35 viruses. Xgenovo assembled significantly longer N50 than other assemblers. Xgenovo generated similar values for the total length of contigs and the number of contigs. Xgenovo increased the maximum length of contig, except for a dataset with 13 viruses, Xgenovo generated a similar value of maximum length (Genovo = 21101 bp, Xgenovo = 21098 bp). MAP generated the lowest assembly capacity.

All assemblers generated no chimera contig (chimera rate = 0%). Xgenovo generated similar cover rate with the original Genovo, while MAP generated the lowest cover rate. Figure 8 shows the CPU time required. Compared to the original Genovo, Xgenovo decreased CPU time by 29.3% (2177 s) for the dataset with 17 viruses, by 48.6% (4414 s) for the dataset with 30 viruses and by 63.7% (16264 s) for the dataset with 35 viruses.

Experiments on different coverage with uniform distribution

The results for experiments on different coverage with uniform distribution were shown in Table 2. Like datasets of log-normal distribution, Xgenovo generated higher N50 than Genovo and MAP for all datasets, shown in Fig. 9. Compared to the original Genovo, Xgenovo increased N50 by: 36.93% (5694 bp) for datasets with the same coverage 30x;

Table 1 **Experiments on different numbers of species with coverage following log-normal distribution.**

Metagenome datasets	Genovo	Xgenovo	MAP
13 viruses		$\beta = 0.3\,\alpha$	
N50 (bp)	8790	11263	399
Total length (bp)	111777	112059	1033790
Max length (bp)/# Contig	21101/13	21098/12	749/2895
Cover rate (%)/Chimera rate (%)	91.15/0	91.15/0	74.05/0
CPU time (s)/Memory (GB)	2365/0.99	2393/1.009	10481/6.053
17 viruses		$\beta = 0.3\,\alpha$	
N50 (bp)	35308	42510	417
Total length (bp)	382890	382183	1348420
Max length (bp)/# Contig	145725/22	168835/17	872/3540
Cover rate (%)/Chimera rate (%)	96.87/0	96.87/0	33.59/0
CPU Time (s)/Memory (GB)	7423/1.301	5246/1.242	4213/6.445
30 viruses		$\beta = 0.3\,\alpha$	
N50 (bp)	16068	35281	258
Total length (bp)	470033	468819	305091
Max length (bp)/# Contig	84444/93	168713/80	897/1050
Cover rate (%)/Chimera rate (%)	95.58/0	95.57/0	37.59/0
CPU time (s)/Memory (GB)	9067/1.372	4653/1.312	28240/10.637
35 viruses		$\beta = 0.3\,\alpha$	
N50 (bp)	11993	21105	259
Total length (bp)	530538	535103	321620
Max length (bp)/# Contig	155979/61	168736/74	722/1064
Cover rate (%)/Chimera rate (%)	97.01	97	39.82
CPU time (s)/Memory (GB)	25504/1.443	9240/1.411	17131/7.598

19.04% (2464 bp) for datasets with the same coverage 40x; 36.77% (5674 bp) for datasets with the same coverage 50x and 36.83% (5676 bp) for datasets with the same coverage 60x. Xgenovo generated similar values for the total length of contigs and the number of contigs. Xgenovo increased maximum length of contig; except for datasets with the same coverage 30x, Xgenovo generated similar values of maximum length (Genovo = 167396 bp, Xgenovo = 163305 bp). Like with log-normal distribution, MAP generated the lowest assembly capacity.

All assemblers generated 0 for chimera rate. Xgenovo generated a similar cover rate with the original Genovo, while MAP generated the lowest cover rate. Figure 10 shows the CPU time required. Compared to the original Genovo, Xgenovo decreased CPU time by 12.45% (448 s) for datasets with the same coverage 30x and by 22.60% (1619 s) for datasets with the same coverage 40x.

Experiments on very complex datasets

To compare the performance between Xgenovo and MAP more rigorously, we generated very complex datasets consisting of 50 viruses, 60 viruses, 90 viruses and 115 viruses, both

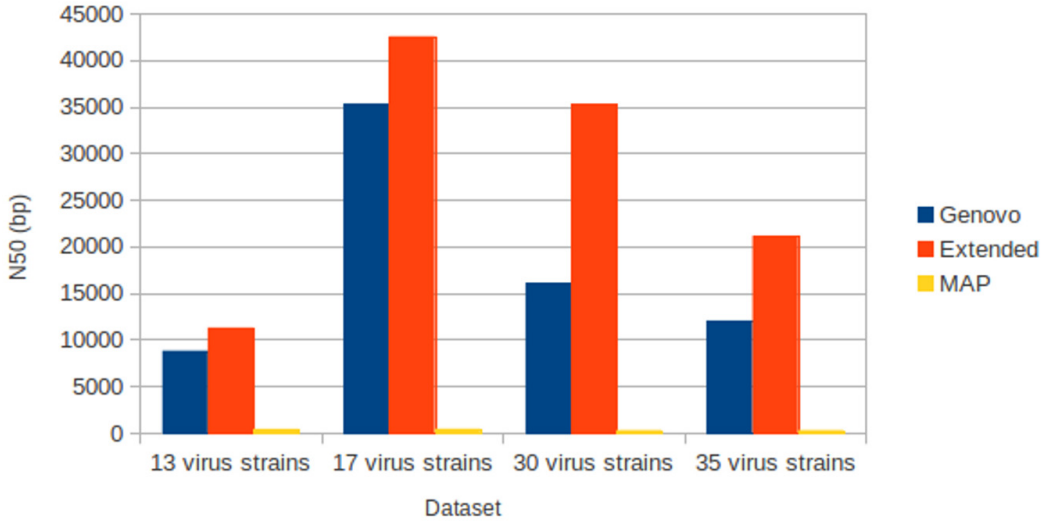

Figure 7 N50 for experiments on different numbers of species with coverage following log-normal distribution. Xgenovo generated the highest N50. The *x*-axis is the name of the dataset and the *y* axis is the N50 (bp).

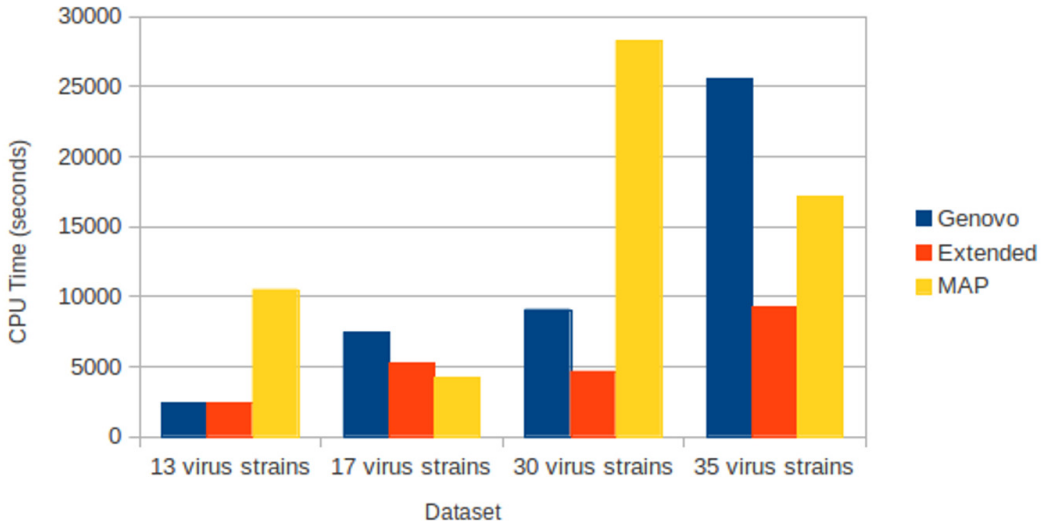

Figure 8 CPU time required for experiments on different numbers of species with coverage following log-normal distribution. The *x*-axis is the name of the dataset and the *y*-axis is the CPU time required (seconds).

for log-normal and uniform distribution. The results were shown in Table 3. Xgenovo generated much higher N50 than MAP for all datasets. Xgenovo generated from 7 times N50 than MAP (for the dataset with 115 viruses with log-normal distribution) until 337 times N50 than MAP (for the dataset of 50 viruses with uniform distribution). For the dataset with 115 viruses with log-normal distribution, MAP generated N50 = 318 bp

Table 2 Experiments on different coverage with uniform distribution.

Metagenome datasets	Genovo	Xgenovo	MAP
Same coverage 30x		$\beta = 0.3\ \alpha$	
N50 (bp)	15415	21109	256
Total length (bp)	533648	534797	210539
Max length (bp)/# Contig	167396/35	163305/36	482/814
Cover rate (%)/Chimera rate (%)	97.57/0	97.57/0	28.19/0
CPU time (s)/Memory (GB)	3597/1.457	3149/1.459	7128/4.818
Same coverage 40x		$\beta = 0.1\ \alpha$	
N50 (bp)	12937	15401	256
Total length (bp)	535585	534993	212998
Max length (bp)/# Contig	84733/40	160954/37	480/824
Cover rate (%)/Chimera rate (%)	97.57/0	97.58/0	27.72/0
CPU time (s)/Memory (GB)	7158/1.479	5539/1.439	7679/4.862
Same coverage 50x		$\beta = 0.1\ \alpha$	
N50 (bp)	15429	21103	256
Total length (bp)	534353	535576	212832
Max length (bp)/# Contig	157303/36	169161/35	483/823
Cover rate (%)/Chimera rate (%)	97.56/0	97.57/0	27.87/0
CPU time (s)/Memory (GB)	3267/1.439	3580/1.441	7173/4.818
Same coverage 60x		$\beta = 0.1\ \alpha$	
N50 (bp)	15410	21086	256
Total length (bp)	534771	535626	218786
Max length (bp)/# Contig	146185/37	169024/36	477/849
Cover rate (%)/Chimera rate (%)	97.58/0	97.58/0	29.27/0
CPU time (s)/Memory (GB)	6069/1.44	6072/1.418	7378/4.812

while Xgenovo generated N50 = 2457 bp. For the dataset with 50 viruses with uniform distribution, MAP generated N50 = 288 bp while Xgenovo generated N50 = 97184 bp. MAP generated a lower chimera rate than Xgenovo but MAP generated very low N50, no more than 392 bp for all datasets while the length of the read is 250 bp. The results showed that Xgenovo outperforms MAP even for very complex datasets.

Discussion

For datasets of log-normal and uniform datasets, compared to Genovo, Xgenovo successfully assembled longer N50. Longer contigs can help extract more information from the reads leading to the discovery of more genes and better functional annotation (*Meyer et al., 2009*). When the N50 score is longer, more complete protein-coding genes are predicted (*Namiki et al., 2012*). Xgenovo also successfully generated higher maximum length of contig in most datasets. These results mean that Xgenovo can increase the assembly capacity. Although it increased the assembly capacity, Xgenovo maintained assembly quality by generating a competitive cover rate and chimera rate value. Compared to MAP, Xgenovo also generated a much higher N50. MAP generated low value for both assembly capacity and quality. For metagenomic datasets which are at low taxonomic level,

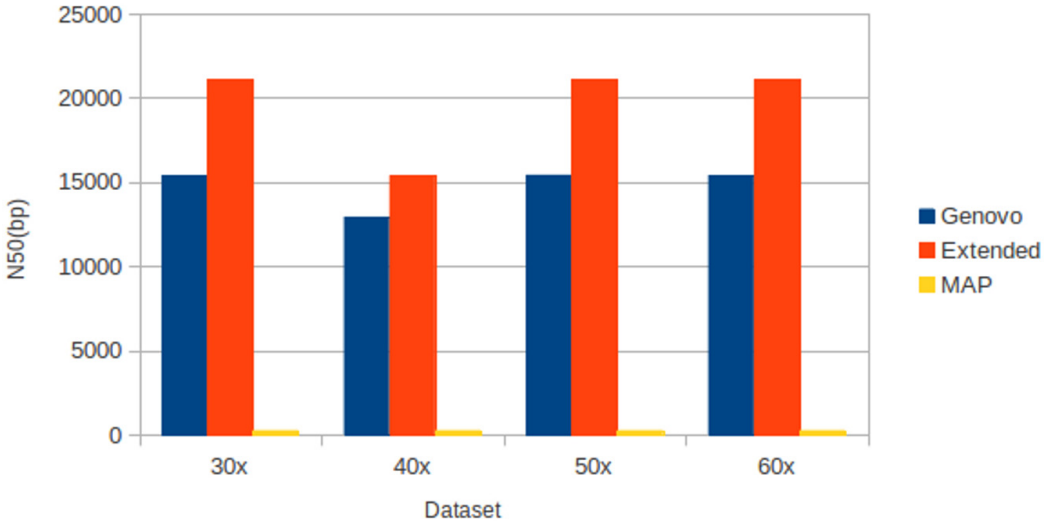

Figure 9 N50 for experiments on different coverages with uniform distribution. Xgenovo generated the highest N50. The *x* axis is the name of the dataset and the *y* axis is the N50 (bp).

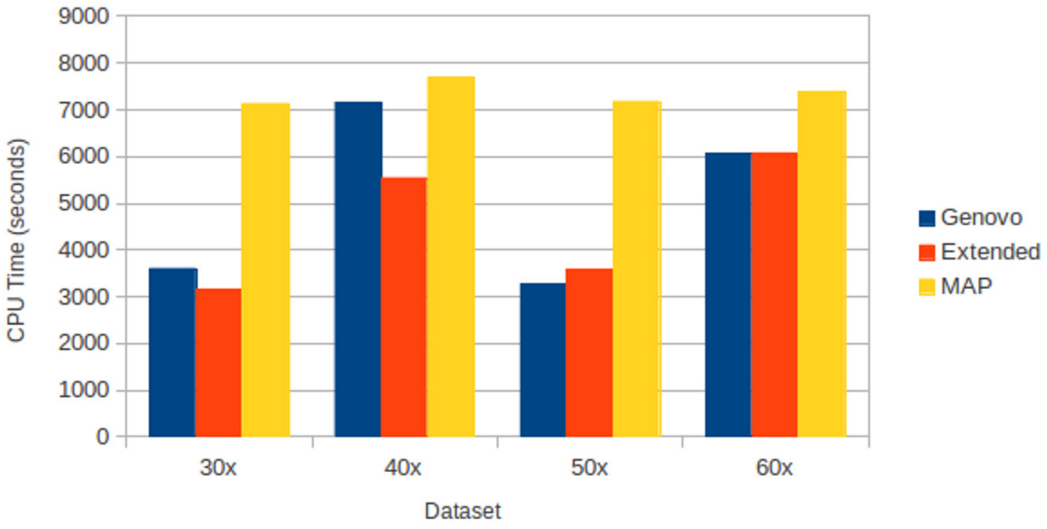

Figure 10 CPU time required for experiments on different coverages with uniform distribution. The *x*-axis is the name of the dataset and the *y*-axis is the CPU time required (seconds).

the genomes become more similar and share more reads with each other. MAP uses an improved OLC (Overlap/Layout/Consensus) strategy to integrate mate pair information which treats a read as a node, therefore the more similar the genomes, the more complex the graph. It might be a reason why MAP generated low performance.

Aside from the improved assembly performance, Xgenovo demonstrated the potential to decrease the computational cost. As explained in the previous section, Genovo uses iterative procedures to discover appropriate assemblies. The main iterative procedure is read mapping. This procedure updates the position of the read in the coordinate system

Table 3 Experiments on very complex datasets.

Metagenome datasets	Xgenovo	MAP
50 Viruses log-normal	$\beta = 0.5\,\alpha$	
N50 (bp)	28903	278
Total length (bp)	1761195	5396779
Max length (bp)/# Contig	110113/502	1479/17966
Cover rate (%)/Chimera rate (%)	94.84/0.93	92.75/0
CPU time (s)/Memory (GB)	39738/2.566	52449/7.906
50 Viruses uniform	$\beta = 0.5\,\alpha$	
N50 (bp)	97184	288
Total length (bp)	1794418	5293906
Max length (bp)/# Contig	194028/123	1123/17519
Cover rate (%)/Chimera rate (%)	96.99/6.07	97.1/0
CPU time (s)/Memory (GB)	41324/2.761	25193/4.147
60 Viruses log-normal	$\beta = 0.1\,\alpha$	
N50 (bp)	10636	293
Total length (bp)	2362232	5478753
Max length (bp)/# Contig	108811/1120	1633/17258
Cover rate (%)/Chimera rate (%)	92.38/1.19	87.59/0
CPU time (s)/Memory (GB)	43206/3.228	16502/3.625
60 Viruses uniform	$\beta = 0.3\,\alpha$	
N50 (bp)	25099	301
Total length (bp)	2840986	5283695
Max length (bp)/# Contig	106273/346	1476/16628
Cover rate (%)/Chimera rate (%)	96.77/1.06	24.44/0.09
CPU time (s)/Memory (GB)	43786/3.618	14478/3.232
90 Viruses log-normal	$\beta = 0.3\,\alpha$	
N50 (bp)	2916	303
Total length (bp)	3003232	5622001
Max length (bp)/# Contig	126010/2640	1864/16890
Cover rate (%)/Chimera rate (%)	85.14/0.63	77.52/0.02
CPU time (s)/Memory (GB)	60162/ 4.27	21104/3.75
90 Viruses uniform	$\beta = 0.5\,\alpha$	
N50 (bp)	6480	362
Total length (bp)	3631520	5728435
Max length (bp)/# Contig	25751/1507	1785/15562
Cover rate (%)/Chimera rate (%)	96.03/1.39	90.85/0.01
CPU time (s)/Memory (GB)	56635/4.27	11656/2.76
115 Viruses log-normal	$\beta = 0.5\,\alpha$	
N50 (bp)	2457	318
Total length (bp)	3437804	5689659
Max length (bp)/# Contig	128459/3410	1806/16620
Cover rate (%)/Chimera rate (%)	81.33/0.84	73.93/0.01
CPU time (s)/Memory (GB)	68789/4.96	10480/ 3.42
115 Viruses uniform	$\beta = 0.1\,\alpha$	
N50 (bp)	4264	392
Total length (bp)	4518801	6269732
Max length (bp)/# Contig	27417/2720	1956/16104
Cover rate (%)/Chimera rate (%)	96.07/0.59	88.07/0.02
CPU time (s)/Memory (GB)	72487/5.29	8137/2.23

of contigs and offsets. This procedure samples the contig of a read utilizing CRP and samples the location of read in the contig utilizing symmetric geometric distribution. This procedure requires the highest computational cost of procedures in Genovo. A read will be resampled if its mapping location in the contig contains some problematic spots. A problematic spot is defined as a spot having supported reads ≤ 2, a spot in the edge of a contig, or a spot which doesn't have a supported read before or after it. If a read doesn't have any problematic spots, the read will not be resampled. In Xgenovo, CRP and the symmetric geometric distribution are modified so that a read sampled in the appropriate location has higher probability which means that a read will be mapped in the correct location. If a read is mapped in the correct location, it contains fewer problematic spots and doesn't need to be resampled. That is why it's possible for Xgenovo to decrease the computational time in the same number of iterations.

CONCLUSION

We successfully extended Genovo by incorporating paired-end information so that it generates higher quality assemblies with paired end reads by modifying Genovo in determining the location of a read in the coordinate system of the contig and the offset (the beginning of the read). Unlike other assemblers (*Koren, Treangen & Pop, 2011*; *Li et al., 2010*; *Namiki et al., 2012*; *Peng et al., 2012*; *Zerbino & Birney, 2008*; *Zerbino et al., 2009*) which use paired end information to generate scaffolds, we attempted to increase the assembly performance without the aim of generating scaffold but attempted to map reads to the contigs in the correct location. Xgenovo successfully generated longer N50 than the original Genovo and the recently proposed matagenome assembler for 454 reads, MAP while maintaining the assembly quality for simulated metagenomic datasets with species coverage following uniform and log-normal distribution even for very complex dataset. Xgenovo also demonstrated the potential to decrease the computational cost. It means that our strategy worked well.

Genovo is the only metagenomic assembler that uses a generative probabilistic model. Unlike the other methods, Genovo assembles all reads without discarding any reads. This strategy avoids filtering out any low coverage genomes, hence hopefully is able to extract more information from metagenomic data in order to generate better assembly results. The consequence is high computational cost. We have improved Genovo by incorporating paired end information and demonstrate that it can reduce computational cost. Short reads, for example Illumina reads, have been gaining popularity, even for metagenomic studies (*Hiatt et al., 2010*). We are going to continue our research and extend our method for short read data in order to generate high assembly accuracy and capacity with reliable computational cost. Current metagenomic assemblers for short read data (Metavelvet, MetaIDBA and IDBA-UD) use the De Bruijn graph approach. Therefore, the implementation of a probabilistic model for short read data with high assembly performances and consistent computational cost is a potential area of research.

ADDITIONAL INFORMATION AND DECLARATIONS

Funding

This work was supported by Grant-in-Aid for KAKENHI (Grant-in-Aid for Scientific Research) on Innovative Areas and Scientific Research (A) from the Ministry of Education, Culture, Sports, Science and Technology of Japan. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Grant Disclosures

The following grant information was disclosed by the authors:
Grant-in-Aid for KAKENHI (Grant-in-Aid for Scientific Research) on Innovative Areas: No. 221S0002.
Ministry of Education, Culture, Sports, Science and Technology of Japan: Scientific Research (A) No. 23241066.

Competing Interests

The authors declare that they have no competing interests.

Author Contributions

- Afiahayati conceived and designed the experiments, performed the experiments, analyzed the data, contributed reagents/materials/analysis tools, wrote the paper.
- Kengo Sato made suggestions in developing the algorithm and performing experiments.
- Yasubumi Sakakibara wrote the paper.

Data Deposition

The following information was supplied regarding the deposition of related data:
http://xgenovo.dna.bio.keio.ac.jp/download.

REFERENCES

Aldous DJ. 1985. Exchangeability and related topics. In: Hennequin PL, ed. *cole d't de probabilits de Saint Flour XIII 1983*, *Lecture Notes in Mathematics,* 1117. Berlin: Springer, 1198.

Chen K, Pachter L. 2005. Bioinformatics for whole-genome shotgun sequencing of microbial communities. *PLoS Computational Biology* **1(2)**:e24 DOI 10.1371/journal.pcbi.0010024.

Degroot MH, Schervish MJ. 2011. *Probability and statistics*. Boston: Addison-Wesley Publishing.

Hiatt JB, Patwardhan RP, Turner EH, Lee C, Shendure J. 2010. Parallel, tag-directed assembly of locally derived short sequence reads. *Nature Methods* **7(2)**:199–122 DOI 10.1038/nmeth.1416.

Johnson M. Chinese Restaurants Process. *Available at http://cog.brown.edu/~mj/classes/cg168/slides/ChineseRestaurants.pdf* (accessed 10 May 2012).

Koren S, Treangen TJ, Pop M. 2011. Bambus 2: scaffolding metagenomes. *Bioinformatics* **27(21)**: 2964–2971 DOI 10.1093/bioinformatics/btr520.

Lai B, Ding R, Li Y, Duan L, Zhu H. 2012. A de novo metagenomic assembly program for shotgun dna reads. *Bioinformatics* **28(11)**:1455–1462 DOI 10.1093/bioinformatics/bts162.

Laserson J, Jojic V, Koller D. 2011. Genovo: de novo assembly for metagenomes. *Journal of Computational Biology* **18(3)**:429–443 DOI 10.1089/cmb.2010.0244.

Lasken RS, Stockwell TB. 2007. Mecanishm of chimera formation during the multiple displacement amplification reaction. *BMC Biotechnology* **7**:19 DOI 10.1186/1472-6750-7-19.

Li R, Zhu H, Ruan J, Qian W, Fang X, Shi Z, Li Y, Li S, Shan G, Kristiansen K, Li S, Yang H, Wang J, Wang J. 2010. De novo assembly of human genomes with massively parallel short read sequencing. *Genome Research* **20**:265–272 DOI 10.1101/gr.097261.109.

Margulies M, Egholm M, Altman WE, Attiya S, Bader JS, Bemben LA, Berka J, Braverman MS, Chen YJ, Chen Z, Dewell SB, Du L, Fierro JM, Gomes XV, Godwin BC, He W, Helgesen S, Ho CH, Irzyk GP, Jando SC, Alenquer MLI, Jarvie TP, Jirage KB, Kim JB, Knight JR, Lanza JR, Leamon JH, Lefkowitz SM, Lei M, Li J, Lohman KL, Lu H, Makhijani VB, McDade KE, McKenna MP, Myers EW, Nickerson E, Nobile JR, Plant R, Puc BP, Ronan MT, Roth GT, Sarkis GJ, Simons JF, Simpsons JW, Srinivasan M, Tartaro KR, Tomasz A, Vogt KA, Volkmer GA, Wang SH, Wang Y, Weiner MP, Yu P, Begley RF, Rothberg JM. 2005. Genome sequencing in microfabricated high-density picolitre reactors. *Nature* **437**:376–380 DOI 10.1038/nature03959.

Mavromatis K, Ivanova N, Barry K, Shapiro H, Goltsman E, McHardy AC, Rigoutsos I, Salamov A, Korzeniewski F, Land M, Lapidus A, Grigoriev I, Richardson P, Hugenholtz P, Kyrpides NC. 2007. Use of simulated data sets to evaluate the fidelity of metagenomic processing methods. *Nature Methods* **4(6)**:495–500 DOI 10.1038/nmeth1043.

Mende DR, Waller AS, Sunagawa S, Jrvelin AI, Chan MM, Arumugam M, Raes J, Bork P. 2012. Assessment of metagenomic assembly using simulated next generation sequencing data. *PLoS ONE* **7(2)**:e31386 DOI 10.1371/journal.pone.0031386.

Meyer E, Aglyamova GV, Wang S, Buchanan-Carter J, Abrego D, Colbourne JK, Willis BL, Matz MV. 2009. Sequencing and de novo analysis of a coral larval transcriptome using 454 gsflx. *BMC Genomics* **10**:219 DOI 10.1186/1471-2164-10-219.

Miller JR, Delcher AL, Koren S, Venter E, Walenz BP, Brownley A, Johnson J, Li K, Mobarry C, Sutton G. 2008. Aggressive assembly of pyrosequencing reads with mates. *Bioinformatics* **24(24)**:2818–2824 DOI 10.1093/bioinformatics/btn548.

Myers EW, Sutton GG, Delcher AL, Dew IM, Fasulo DP, Flanigan MJ, Kravitz SA, Mobarry CM, Reinert KHJ, Remington KA, Anson EL, Bolanos RA, Chou HH, Jordan CM, Halpern AL, Lonardi S, Beasley EM, Brandon RC, Chen L, Dunn PJ, Lai Z, Liang Y, Nusskern DR, Zhan M, Zhang Q, Zheng X, Rubin GM, Adams MD, Venter JC. 2000. A Whole-Genome Assembly of Drosophila. *Science* **287(5461)**:2196–2204 DOI 10.1126/science.287.5461.2196.

Nagarajan N, Pop M. 2013. Sequence assembly demystified. *Nature Reviews Genetics* **14**:157–167 DOI 10.1038/nrg3367.

Namiki T, Hachiya T, Tanaka H, Sakakibara Y. 2012. Metavelvet: an extension of velvet assembler to de novo metagenome assembly from short sequence reads. *Nucleic Acids Research* **40(20)**:e155 DOI 10.1093/nar/gks678.

Peng Y, Leung HCM, Yiu SM, Chin FYL. 2010. IDBA a practical iterative de bruijn graph de novo assembler. In: Berger B, ed. *Research in computational molecular biology*, *Lecture Notes in Computer Science*. Berlin: Springer, 426–440.

Peng Y, Leung HCM, Yiu SM, Chin FYL. 2011. Meta-IDBA: a de novo assembler for metagenomic data. *Bioinformatics* **27(13)**:i94–101 DOI 10.1093/bioinformatics/btr216.

Peng Y, Leung HC, Yiu SM, Chin FY. 2012. IDBA-UD: a de novo assembler for single-cell and metagenomic sequencing data with highly uneven depth. *Bioinformatics* **28(11)**:1420–1428 DOI 10.1093/bioinformatics/bts174.

Pigmatelli M, Moya A. 2011. Evaluating the fidelity of de novo short read metagenomic assembly using simulated data. *PLoS ONE* **6(5)**:e19984 DOI 10.1371/journal.pone.0019984.

Qin J, Li R, Raes J, Arumugam M, Burgdorf KS, Manichanh C, Nielsen T, Pons N, Levenez F, Yamada T, Mende DR, Li J, Xu J, Li S, Li D, Cao J, Wang B, Liang H, Zheng H, Xie Y, Tap J, Lepage P, Bertalan M, Batto JM, Hansen T, Le Paslier D, Linneberg A, Nielsen HB, Pelletier E, Renault P, Sicheritz-Ponten T, Turner K, Zhu H, Yu C, Li S, Jian M, Zhou Y, Li Y, Zhang X, Li S, Qin N, Yang H, Wang J, Brunak S, Dore J, Guarner F, Kristiansen K, Pedersen O, Parkhill J, Weissenbach J, MetaHIT Consortium, Bork P, Ehrlich SD, Wang J. 2010. A human gut microbial gene catalogue established by metagenomic sequencing. *Nature* **464(7285)**:59–65 DOI 10.1038/nature08821.

Richter DC, Ott F, Auch AF, Schmid R, Huson DH. 2008. Metasim a sequencing simulator for genomics and metagenomics. *PLoS ONE* **3(10)**:e3373 DOI 10.1371/journal.pone.0003373.

Scholz MB, Lo CC, Chain PS. 2012. Next generation sequencing and bioinformatic bottlenecks: the current state of metagenomic data analysis. *Current Opinion in Biotechnology* **23(1)**:9–15 DOI 10.1016/j.copbio.2011.11.013.

Unterseher M, Jumpponen A, Opik M, Tedersoo L, Moora M, Dormann CF, Schnittler M. 2011. Species abundance distributions and richness estimations in fungal metagenomics lessons learned from community ecology. *Molecular Ecology* **20**:275–285 DOI 10.1111/j.1365-294X.2010.04948.x.

Zerbino D, Birney E. 2008. Velvet: algorithms for de novo short read assembly using de bruijn graphs. *Genome Research* **18**:821–829 DOI 10.1101/gr.074492.107.

Zerbino DR, McEwen GK, Margulies EH, Birney E. 2009. Pebble and rock band: heuristic resolution of repeats and scaffolding in the velvet short-read de novo assembler. *PLoS ONE* **4**:e8407 DOI 10.1371/journal.pone.0008407.