Evaluation of Short Read Metagenomic Assembly

Anveshi Charuvaka
Dept. of Computer Science
George Mason University
Email: acharuva@gmu.edu

Huzefa Rangwala
Dept. of Computer Science
George Mason University
Email: rangwala@cs.gmu.edu

Abstract—Assembling short reads obtained from community samples using next-generation sequencing technologies is challenging due to several reasons. In this study we assess the performance of a state-of-the-art Eulerian-path based assembler on a series of simulated dataset with varying complexities. We evaluate the feasibility of metagenomic assembly with reads restricted to length 36 base pairs, obtained from the Solexa/Illumina platform.

We developed a pipeline to evaluate the quality of assembly based on contig length statistics and accuracy. We studied the effect of overlap parameters used for the metagenomic assembly and developed a clustering solution to pool the contigs obtained from different runs of the assembly algorithm which allowed us to obtain longer contigs from different runs. We also computed an entropy/impurity metric to assess the degree of chimericity in the assembled contigs. We also compared the metagenomic assemblies to the best possible solution that could be obtained by assembling individual source genomes. Our results show that accuracy was better than expected for the metagenomic samples with a few dominant organisms and was especially poor in samples containing many closely related strains.

I. INTRODUCTION

Advances in sequencing technologies have equipped researchers with the ability to sequence collective genomes of entire microbial communities, commonly referred to as metagenomics, in an inexpensive and high-throughput manner. Microbes are omnipresent within the human body and environments across the world. As such, characterizing and understanding their roles is crucial for improving human health and the environment. Metagenomics provides an unbiased view of the diversity and biological potential of microbial communities [1] and analysis of community samples from several different microbial environments such as sea [2], soil and human body [3] has provided some key insights into the understandings of these microbial communities.

One of the major challenges related to metagenomic processing is the assembly of short reads obtained from community samples. Although no tools have been developed specifically to address metagenomic assembly, computational tools exist for related problems such as phylogenetic classification of reads, unsupervised clustering or binning, comparative metagenomic analysis, and gene prediction. We refer the reader to [4] for an excellent review of computational challenges and available tools for metagenomics. Due to the lack of specific assemblers to handle metagenomes, researchers continue to use assemblers originally developed for whole genome assembly. Genome assembly from sequencing reads is a challenging task due to sequencing errors, presence of repeats and non-uniform coverage of sequences due to sequencing biases. Coupled with additional problems of metagenomic samples such as unequal sampling depths of constituent organisms and presence of multiple related strains, assembly becomes even more challenging. Consequently, contigs from metagenomes may contain a greater percentage of mis-assemblies [5].

In this paper we assess the performance of a state-of-the-art Eulerian-path based sequence assembler on simulated metagenomic datasets. We used a read length of 36 base pairs (bp), as produced by the Solexa/Illumina sequencing technology. The datasets were meant to reflect the different complexities of real metagenomic samples [6]. They included, a low complexity dataset with one dominant organism, a high complexity dataset with no dominant organism and a medium complexity dataset having a few dominant organisms. We also created a dataset containing different strains of the same organism to measure the extent of co-assembly when reads from very similar organisms are used. Since the metagenomic read datasets are voluminous, we used a parallel sequence assembly algorithm (ABYSS [7]) which can be deployed easily on a commodity Beowulf Linux cluster.

The assembled contigs were evaluated based on several quality measures for contig length and assembly accuracy. To improve the quality of the contigs, we clustered the results of different parameter runs of the assembler. We used efficient local alignment to quickly and accurately map the assembled contigs to the input source genomes. We also used a short read mapping algorithm to align the input reads to the assembled contigs to compute the homogeneity of the assembled contigs using entropy as a metric. Finally, we assessed the coverage of the source genomes by the produced contigs.

Short-read assembly of metagenomes performed better than our initial expectation in some aspects such as accuracy of the contigs and coverage of the source genomes. However,
fragmentation of the contigs was more severe in metagenomic datasets than in the isolate assemblies. The assembly of a smaller dataset consisting of reads from 30 EColi strains showed that the contigs obtainable through co-assembly of related strains are considerably shorter than those generated using isolate assemblies. We also observed that by clustering results from assembly runs for different k-mer size values of de Bruijn graph we were able to obtain a greater number of longer contigs (as optimal contigs are distributed across the k-mer space).

II. RELATED WORK

In our current work, we have estimated the extent of problems associated with the assembly of short reads obtained from next generation sequencing Solexa platform for metagenomic samples. A similar study by Mavromatis et al. [6] produced three simulated metagenomic datasets representing microbial communities of different complexities using reads obtained from Sanger-based sequencing. They used these datasets for benchmarking various metagenomic processing methods. One of the focuses of their study was estimating the chimericity in assembling reads obtained from Sanger sequencing using Overlap-Layout-Consensus (OLC) based assemblers commonly used for isolate genome assembly. Another simulation study by Wommack et al. [8] evaluated simulated NGS short reads from different metagenomic samples for taxonomic and functional annotation. As more and more metagenomic projects have started to tap into the potential of NGS, we felt the need for a similar simulation study to evaluate short read assemblers. The work of Pop [5] provides a good overview of OLC and Eulerian assembly paradigms and addresses some of the challenges associated with short read assembly. Since the next generation sequencing allows the samples to be sequenced at a greater depth, we considered considerably larger datasets. Several researchers have studied the performance of NGS short reads and paired-end short reads for individual genome assembly [9]–[11]. Recently, Kingsford et al. [12] performed a theoretical analysis of Eulerian-path based approaches to survey the repeat structure of individual prokaryotic genomes.

III. METHODS

A. Datasets

To study the extent of errors in metagenomic assemblies in comparison to single genome assembly, we performed a set of experiments on simulated datasets. Although, simulated datasets do not completely capture the characteristics of real metagenomes [13], simulation studies do provide some insight into the feasibility of assembly of short read metagenomic samples.

We created our simulated datasets using Metasim [14]. It is a sequencing simulation tool for generating synthetic metagenomic datasets using a collection of complete genomic sequences. Metasim provides options for controlling various simulation parameters such as sequencing platform, read length, sequencing depth of individual sequences, error rate, and error distribution. We generated reads of length 36 bp using the default empirical error model of Metasim, which simulates the reads produced by Solexa sequencing technology. The bacterial sequences for generating the reads were taken from the completely assembled bacterial genomes from NCBI [http://www.ncbi.nlm.nih.gov/]

![Figure 1](image.png)
ditional information related to the datasets has been made available at the supplementary website.  

We constructed a fourth dataset, **EcoliStrains** consisting of 10 million reads sampled uniformly from 30 different strains of *Escherichia coli*. The coverage of each strain was approximately 2.3x. This dataset was constructed to study the extent of co-assembly when reads from very similar organisms are assembled together.

### B. Assembly

Due to the high computational requirements for the assembly of our metagenomic datasets, we used ABYSS [7] assembler which can run parallel assembly on a cluster of commodity computers. We assembled all of our datasets using ABYSS, with read length of 36 bp and varied the k-mer size parameter of ABYSS’s distributed de Bruijn Graph between 21 and 33 (in increments of two) to obtain different assemblies. In the presence of sequencing errors the optimal k-mer size for Eulerian path based assemblers is determined by the coverage of source sequences. For high coverage, values close to read length produce longer contigs. Similarly, if the percentage of sequencing errors is high, optimal results are obtained by decreasing the k-mer size. All the assembly jobs were run using 32 cluster nodes. We filtered out contigs shorter than 50 bp from the final assemblies.

Due to the presence of sequencing errors and repeat regions in the source genome, assemblies are usually not completely error free, even in the context of a single genome. We assembled the reads from the individual source genomes by separating them first. We ran ABYSS on each genome (reads) within the metagenomic sample individually and combined all the produced contigs. In this study, the contigs produced in this manner are referred to as isolate assembly and provides us a comparative baseline to the metagenomic assembly.

### C. Clustering Assembled Contigs

We observed that the contigs of optimal length were distributed across the k-mer space. Therefore, we pooled the assembled contigs from different contig sets (obtained using different k-mer values) and clustered them to remove duplicate or suboptimal contigs which were contained in another longer contig. We clustered the contigs using Cd-hit [15], which uses a greedy incremental algorithm. The first cluster is formed using the longest sequence as the cluster representative, and the remaining sequences are compared to it in decreasing order of their lengths. If a sequence matches to one of the cluster representatives with sufficient accuracy, then it is placed in that cluster. Otherwise, a new cluster is formed with the unmatched sequence as the cluster representative. Instead of performing the actual alignment, Cd-hit uses a short word filtering algorithm to compute sequence similarity, therefore, it achieves significant speed-up compared to alignment based clustering tools. We clustered our assemblies using a similarity threshold of 96% and a word size of 8 bases (recommended for clustering with high similarity).

### D. Contig Alignment to Reference

To estimate the assembly accuracy we aligned the contigs to the source genomes. Accurate contigs are expected to match at least one source sequence with high accuracy. Therefore, to speed up the alignment process we used NUCMER pipeline of MUMMER [16]. NUCMER uses a suffix tree based string matching algorithm to search for exact matches and extends these matches using a dynamic programming based alignment and is considerably faster than BLAST. We set the parameters for exact match size to 15 and minimum cluster size to 30 and collected all possible matches, so that the sensitivity is not significantly affected. We calculated the contig accuracy by normalizing the accuracy of the local alignment (localAcc) produced by NUCMER using the ratio of length of the alignment (alignLen) to the length of the contig (contigLen), as given by (1).

\[
\text{contig accuracy} = \text{localAcc} \times \frac{\text{alignLen}}{\text{contigLen}} \quad (1)
\]

Some contigs, especially the shorter ones, produced multiple alignments, either to the same or to a different genome. Therefore, we used the best accuracy among all the alignments as the contig’s assembly accuracy. For contig coverage calculations (discussed later on) we consider only the contigs that were assembled with a threshold accuracy of at least 95%. We would like to note that this threshold is rather arbitrary and in many cases, the acceptable accuracy threshold would be dependent on particular application. Our choice was inspired by a similar threshold used by the authors of ABYSS [7] for evaluating the assembly accuracy for isolate genomes.

### E. Contig Homogeneity Calculations

We estimated the homogeneity of contigs by observing the source genome of reads used to assemble a contig. This was done by performing a read-to-contig alignment using a fast short read aligner. We used BWA [17], which performs a backward search with burrows wheeler transform and efficiently aligns short reads against reference sequences. In our case, the references consisted of the set of contigs. Each read was assigned to the contig to which BWA reported the best match.

Using the counts of reads from each source sequence mapped to a given contig, we calculated the entropy of the contigs as shown in (2).

\[
\text{entropy} = - \sum_i p_i \log(p_i) \quad (2)
\]

2http://www.cs.gmu.edu/~mlbio/supplements/short-read-mgs
Figure 2. Contig length distribution for different k-mer sizes, clustered contigs and comparison of metagenomic and isolate assemblies.

where $p_i$ is the fraction of total reads coming from source genome $i$. At different phylogenetic levels, organisms generally display a greater sequence similarity within their group when compared with the organisms belonging to a different group. Due to this sequence similarity, the assemblers are more likely to make the mistake of mis-assembling reads belonging to the same phylogenetic class. We also compute the entropy at two higher phylogenetic levels, genus and phylum, in addition to the entropy at sequence and strain level, to see if there is a significant decrease in entropy at higher phylogenetic levels.

The need for a short-read aligner arises because, for Eulerian path based assemblers, it is difficult to determine the actual read composition of the contigs. The input reads are not used directly but are broken down into smaller k-mers and the original read information is lost. We also computed an impurity metric but do not report the details here because of its redundancy with the entropy measure.

F. Source Coverage Ratio

For different assemblies generated by varying the values of assembly parameter $k$, we calculated the extent to which the source sequences are represented by contigs. This is performed by aligning the contigs to the source sequences. We considered only the accurate alignments of the contigs, i.e. the alignments which accurately cover $\geq 95\%$ of the contig. For each such alignment, we marked all the positions of the source genomes which were part of the alignments. The collection of all such positions of the source genome covered by the contigs, represents the contig coverage of the genome. The contig coverage described here is different from the read coverage which is approximate coverage of the genome from the reads and represents the sequencing depth of the source sequences in the datasets. The contig coverage represents the fraction of the source sequence recovered by the contigs and can be at most 1. We defined this contig coverage to be the source coverage ratio.

Where contigs had multiple accurate alignments, possibly due to repeat regions or shared sequences between genomes, we counted each contig’s contribution for all the alignments. Therefore, our contig mapping to source genomes is not unique, and our source coverage ratio calculation may have over-counted a little. As it is not possible to prefer one particular alignment over another, we believe this is a better
option than randomly choosing a particular alignment of the contig.

IV. RESULTS

We evaluated the metagenomic assembly based on the accuracy of the generated contigs using alignment-based similarity to the source genomes, contig length statistics, and the proportions of the source genomes recovered by the contigs. As the k-mer size of de Bruijn graph plays a crucial role in ABYSS’s assembly, we assembled the datasets at different values of k-mer size and compared the results. We also compared the contig length statistics of metagenomic assemblies to isolate assemblies.

A. Contig Length vs K-mer size

Fig. 2 shows a comparison of the contig lengths to k-mer size across the simLC, simMC and simHC datasets. The horizontal axis represents the contig length, and the vertical axis represents the number of contigs (in log scale) greater than or equal to the given threshold length. The average lengths of the contigs decrease with an increase in the complexity of the dataset. From the Fig. 2 (a) & (c) it can be seen that the optimal value of $k$ to obtain longer contigs changes from 29 for simLC (a single dominant genome at a very high coverage) to 21 and 23 for simHC (does not have any distinctly dominant genomes). As seen from Fig. 2(b), the simMC dataset $k = 25$ seems to produce the longer contigs. The clustered results effectively pool the contigs produced by setting different $k$ values across the different datasets.

Table. I provides additional statistics regarding the assembled contigs including $N_{50}$ (weighted median) value. In general, the number of bases recovered increased with a decrease in k-mer size. It seems that for smaller values of $k$ we are able to assemble more of the low coverage sequences but the contigs tend to be more fragmented.

B. Metagenomic vs Isolate Assembly

As a benchmark for our metagenomic assemblies we separated the reads by their source sequence and performed isolate assemblies. We assembled the reads from each individual sequence separately and combined the final contigs from all the source sequences. We performed the isolate assemblies with different values of $k$ and pooled the results using the clustering algorithm. Fig. 2 (d) compares the length distribution of clustered results form metagenomic and isolate assemblies.

The simHC dataset produced shorter contigs in both isolate as well as metagenomic assemblies. Amongst the simLC and simHC datasets, the performance of simLC was closer to the isolate assemblies, whereas, the simMC metagenomic assembly was far poorer in comparison to its isolated assembly.

C. Contig Alignment Accuracy

Even assemblies of isolate genomes are not completely error free. In the case of metagenomes, the presence of multiple genomes at different coverage depths causes additional problem and the contigs are expected to have more mis-assemblies compared to the contigs from isolate genome assemblies. We compute the contig alignment accuracy (Section section III-D) and report the results for different datasets in Table. I. A threshold accuracy of 95% was used for considering a contig accurate. The assembly accuracies decreased as the k-mer size was decreased and was worst for all datasets at $k=21$. Further, the accuracy of the clustered contigs was lowest, due to the accumulation of errors from
| Dataset          | K   | Total Contigs | Accurate Contigs | % Accurate Contigs | N50 | Total Bases | Bases in Accurate Contigs | % Bases in Accurate Contigs |
|------------------|-----|---------------|------------------|-------------------|-----|-------------|---------------------------|-----------------------------|
| EcoliStrains-10m | C   | 25742         | 25359            | 98.31             | 1223| 998000      | 991374                    | 99.29                       |
|                  | 21  | 24883         | 24709            | 99.3              | 544 | 666096      | 662743                    | 99.51                       |
|                  | 23  | 20550         | 20459            | 99.56             | 847 | 656049      | 654584                    | 99.78                       |
|                  | 25  | 19570         | 19506            | 99.67             | 933 | 637041      | 635678                    | 99.79                       |
|                  | 27  | 17474         | 17422            | 99.7              | 1195| 599591      | 598649                    | 99.84                       |
|                  | 29  | 17338         | 17278            | 99.65             | 925 | 556057     | 555039                    | 99.82                       |
|                  | 31  | 25468         | 25436            | 99.87             | 317 | 523787      | 523375                    | 99.92                       |
| simMC-36m        | C   | 119667        | 112827           | 94.28             | 493 | 34920211    | 34050685                  | 97.51                       |
|                  | 21  | 100852        | 96658            | 97.82             | 566 | 31992027    | 31595445                  | 98.76                       |
|                  | 23  | 183383        | 178586           | 97.38             | 161 | 24663896    | 24173013                  | 99.73                       |
|                  | 25  | 106981        | 106133           | 99.21             | 324 | 24661676    | 24510881                  | 99.39                       |
|                  | 27  | 88466         | 88122            | 99.61             | 419 | 23280192    | 23216974                  | 99.73                       |
|                  | 29  | 78057         | 77825            | 99.68             | 569 | 21119299    | 21078047                  | 99.79                       |
|                  | 31  | 75800         | 75571            | 99.7              | 400 | 18839097    | 1877496                   | 99.67                       |
|                  | 33  | 114336        | 113948           | 99.66             | 168 | 17128489    | 17050171                  | 99.54                       |
| simHC-36m        | C   | 75480         | 55508            | 75.54             | 138 | 10007373    | 7649152                   | 76.44                       |
|                  | 21  | 39196         | 35369            | 90.24             | 131 | 5366108     | 4823423                   | 98.89                       |
|                  | 23  | 51371         | 36693            | 71.43             | 142 | 6923506     | 5037993                   | 72.77                       |
|                  | 25  | 28614         | 25707            | 89.84             | 137 | 4132826    | 3703686                   | 89.62                       |
|                  | 27  | 17418         | 16557            | 95.06             | 122 | 2289332    | 2179104                   | 95.19                       |
|                  | 29  | 9822          | 9524             | 96.97             | 109 | 1184664    | 1149541                   | 97.04                       |
|                  | 31  | 5309          | 5211             | 98.15             | 102 | 603680     | 593152                    | 98.26                       |
|                  | 33  | 3047          | 3005             | 98.62             | 93  | 315736     | 311501                    | 98.66                       |
|                  |     | 1895          | 1885             | 99.47             | 77  | 162625     | 161704                    | 99.43                       |
| simLC-36m        | C   | 119667        | 112827           | 94.28             | 493 | 34920211    | 34050685                  | 97.51                       |
|                  | 21  | 100852        | 96658            | 97.82             | 566 | 31992027    | 31595445                  | 98.76                       |
|                  | 23  | 183383        | 178586           | 97.38             | 161 | 24663896    | 24173013                  | 99.73                       |
|                  | 25  | 106981        | 106133           | 99.21             | 324 | 24661676    | 24510881                  | 99.39                       |
|                  | 27  | 88466         | 88122            | 99.61             | 419 | 23280192    | 23216974                  | 99.73                       |
|                  | 29  | 78057         | 77825            | 99.68             | 569 | 21119299    | 21078047                  | 99.79                       |
|                  | 31  | 75800         | 75571            | 99.7              | 400 | 18839097    | 1877496                   | 99.67                       |
|                  | 33  | 114336        | 113948           | 99.66             | 168 | 17128489    | 17050171                  | 99.54                       |
|                  |     | 156709        | 156272           | 99.72             | 78  | 12688930    | 12646726                  | 99.67                       |

The entropy metric is computed at four levels: (i) sequence, (ii) species, (iii) genus and (iv) phylum, derived from the NCBI taxonomy tree. Majority of the longer contigs were homogeneous. The simHC dataset produces a large number of smaller inhomogeneous contigs due to insufficient coverage of the source sequences. The proportion of inhomogeneous contigs is comparatively lower in the MC and significantly lower in LC datasets. The contigs were more homogeneous at higher phylogenetic levels. Because the genomes which are phylogenetically close together share significant sequence similarity, there is a greater chance of assembling reads belonging to related sequences into the same contig.

E. Coverage of the source sequences

Fig. 5 shows a plot of the source coverage ratio for the clustered contig sets of simLC, simMC, and simHC datasets. The positions of the source sequences in Fig. 5

all the contig sets. This is due to our clustering approach, which tries to retain all the unique sequences. An alternative clustering strategy could be designed that retains only the contigs found in more than one contig sets. This strategy would improve the accuracy results while reducing the total number of bases recovered.

Figure 3 shows a density plot comparing the accuracy of contigs with respect to the contig lengths across the simLC, simMC and simHC datasets. The density plot is a scatter plot but also color codes points based on the number of instances exhibiting the same values of contig length and accuracy. We noticed that a large percentage of the contigs were assembled accurately. The longer contigs are generally more homogeneous at higher phylogenetic levels. Because the genomes which are phylogenetically close together share significant sequence similarity, there is a greater chance of assembling reads belonging to related sequences into the same contig.

D. Contig Homogeneity

In an ideal case of metagenomic assembly, all the reads forming a contig would come from the same source sequence and the entropy as defined in section III-E, will be 0. In metagenomes the probability of co-assembly of reads from related sources is higher. Therefore, to estimate the degree of chimericity, we evaluated the homogeneity of contigs using their read composition.

Fig. 4 shows a plot of contig entropy versus length of the contigs across the datasets of different complexities. The entropy metric is computed at four levels: (i) sequence, (ii) species, (iii) genus and (iv) phylum, derived from the NCBI taxonomy tree. Majority of the longer contigs were homogeneous. The simHC dataset produces a large number of smaller inhomogeneous contigs due to insufficient coverage of the source sequences. The proportion of inhomogeneous contigs is comparatively lower in the MC and significantly lower in LC datasets. The contigs were more homogeneous at higher phylogenetic levels. Because the genomes which are phylogenetically close together share significant sequence similarity, there is a greater chance of assembling reads belonging to related sequences into the same contig.
Figure 4. Contig entropy to measure contig homogeneity computed across different phylogenetic levels.

correspond to those in the read coverage plot in Fig. 1. Due to space constraints, we did not include the plots showing the coverage values for different k-mer sizes, but we summarize the results here. In almost all the assemblies, a high proportion of source genomes sequenced at higher depth was recovered by the contigs. As the value of k-mer size was decreased, more of the genomes sequenced at lower depth were recovered. However, as evident from the contig length distribution plot, Fig. 2, some values of k-mer size tend to be suboptimal length-wise, depending on the complexity of the datasets. Therefore, clustering of the contigs resolves this issue, as the clustered results retain the longer contigs and also the unique contigs representing the low read coverage genomes.

F. Escherichia Strains Co-assembly

Since the collection of DNA sequences for metagenomic experiments does not involve cloning, the reads could come from strains which are highly similar, with very little sequence variation. In this case, even though the effective read coverage of the species is high, due to minor differences in the sequences of the strains, the quality of assembly might not be as good as an isolate genome assembly. To evaluate the performance of co-assembly of reads from related strains, we created the EColiStrains dataset consisting of 10 million reads from 30 strains of Escherichia Coli. For comparing the assembly performance, we created another dataset with the same number of reads from a single strain, E.Coli strain 536 (represents the isolate assembly), and assembled it using the different k-mer size values used for assembly of the strains dataset. For isolate assembly, $k=27$ and 29 produced the longest contigs. Fig. 6 shows a comparison of isolated and metagenomic strains datasets for $k=27$ and 29. The contigs in metagenomic assembly are considerably shorter than the isolate assembly, suggesting a severe performance degradation resulting from the presence of multiple strains. Fig. 7 shows the source coverage ratio of the constituent strains for different k-mer size values. A relatively high percentage of the source sequences was recovered by the contigs. Table. I provides some additional assembly statistics for EColiStrains dataset. The EColiStrains dataset exhibited some of the same general trends as the simLC, simMC, and simHC datasets. But, the variations in the total number of bases and contig accuracies were less pronounced.

Figure 5. Coverage of source sequences from the clustered contigs.

Figure 6. Contig lengths of EcoliStrains dataset and isolate assembly.
V. CONCLUSION

In this paper we have presented the results of assembly and analysis of some simulated metagenomic datasets. Short-read assembly of metagenomes performed better than our initial expectations in some aspects such as accuracy of the contigs and coverage of the source genomes. Although a large fraction of the contigs were assembled accurately, fragmentation of the contigs was more severe in metagenomic datasets when compared to the isolate assemblies. Further, assembling the high complexity dataset was more difficult in comparison to the low complexity datasets as well. The assembly of a smaller dataset consisting of reads from 30 EColi strains showed that the contigs obtainable through co-assembly of related strains are considerably shorter than those generated using isolate assemblies.

We have also observed that by simple clustering of the results from various assembly runs (obtained from different k-mer size values of de Bruijn graph) we are able to obtain a greater number of longer contigs, as optimal contigs are distributed across the k-mer space. However, due to our simple approach towards clustering which retains all unique contigs, most mis-assembled contigs made their way into the clustered results, increasing their error rate. Further improvements in clustering technique may be needed to improve the quality of the clustered results. We are currently assessing the performance of the assembly algorithms for paired-end reads and the insert length size.

ACKNOWLEDGMENT

This work was supported by NSF IIS-0905117 and George Mason seed grants.

REFERENCES

[1] G. W. Tyson and P. Hugenholtz, “Metagenomics,” *Nature Reviews Microbiology*, Sep 2008.

[2] J. Venter, K. Remington, J. Heidelberg, A. Halpern, D. Rusch, J. Eisen, D. Wu, I. Paulsen, K. Nelson, W. Nelson et al., “Environmental genome shotgun sequencing of the Sargasso Sea,” *Science*, vol. 304, no. 5667, p. 66, 2004.

[3] J. Qin et al., “A human gut microbial gene catalogue established by metagenomic sequencing,” *Nature*, vol. 464, no. 7285, pp. 59–65, Mar 2010.

[4] J. Wooley and Y. Ye, “Metagenomics: Facts and Artifacts, and Computational Challenges,” *Journal of Computer Science and Technology*, vol. 25, no. 1, pp. 71–81, 2009.

[5] M. Pop, “ Genome assembly reborn: recent computational challenges,” *Brief Bioinform.*, vol. 10, no. 4, pp. 354–66, Jul 2009.

[6] K. Mavromatis et al., “Use of simulated data sets to evaluate the fidelity of metagenomic processing methods,” *Nat Methods*, vol. 4, no. 6, pp. 495–500, Jun 2007.

[7] J. T. Simpson, K. Wong, S. D. Jackman, J. E. Schein, S. J. M. Jones, and I. Birol, “Abyss: a parallel assembler for short read sequence data,” *Genome Res.*, vol. 19, no. 6, pp. 1117–23, Jun 2009.

[8] K. E. Wommack, J. Bhavsar, and J. Ravel, “Metagenomics: read length matters,” *Appl Environ Microbiol*, vol. 74, no. 5, pp. 1453–63, Mar 2008.

[9] N. Whiteford, N. Haslam, G. Weber, A. Prugel-Bennett, J. W. Essex, P. L. Rouch, M. Bradley, and C. Neylon, “An analysis of the feasibility of short read sequencing,” *Nucl. Acids Res.*, vol. 33, no. 19, pp. e171–, 2005.

[10] M. J. Chaisson, D. Brinza, and P. A. Pevzner, “De novo fragment assembly with short mate-paired reads: Does the read length matter?” *Genome Research*, vol. 19, no. 2, pp. 336–346, 2009.

[11] S. Mitra, M. Schubach, and D. Huson, “Short clones or long clones? a simulation study on the use of paired reads in metagenomics,” *BMC Bioinformatics*, vol. 11, no. Suppl 1, p. S12, 2010.

[12] C. Kingsford, M. Schatz, and M. Pop, “Assembly complexity of prokaryotic genomes using short reads,” *BMC Bioinformatics*, vol. 11, no. 1, p. 21, 2010.

[13] J. Morgan, A. Darling, and J. Eisen, “Metagenomic Sequencing of an In Vitro-Simulated Microbial Community,” *Public Library of Science*, 2010.

[14] D. C. Richter, F. Ott, A. F. Auch, R. Schmid, and D. H. Huson, “Metasim: a sequencing simulator for genomics and metagenomics,” *PLoS One*, vol. 3, no. 10, p. e3373, 2008.

[15] W. Li and A. Godzik, “Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences,” *Bioinformatics*, vol. 22, no. 13, pp. 1658–9, Jul 2006.

[16] A. Delcher, S. Kasif, R. Fleischmann, J. Peterson, O. White, and S. Salzberg, “Alignment of whole genomes mummer,” *Nucleic Acids Research*, vol. 27, no. 11, p. 2369, 1999.

[17] H. Li and R. Durbin, “Fast and accurate long-read alignment with Burrows-Wheeler transform,” *Bioinformatics*, vol. 26, no. 5, p. 589, 2010.