Metagenomics Assemblers Evaluation [Or Whatever Titus suggests :)]

author[1], Sherine Awad [In whatever order and with whoever should be added] [2], author[3,*]
2 Author1 Dept/Program/Center, Institution Name, City, State, Country
3 Same as Titus Departments/Program/Center, Institution Name, City, State, Country
4 Author3 Dept/Program/Center, Institution Name, City, State, Country
∗ E-mail: Corresponding author@institute.edu

Abstract

Author Summary

Introduction

Metagenome is the sequencing of DNA in an environmental sample. While whole genome sequencing (WGS) usually targets one genome, metagenome targets several ones. Here rises the metagenome assembly problem, from the genomic diversity and variable abundance within populations. In this paper, we propose a comparative study for four different assemblers; Velvet [3], SPAdes [5], IDBA-UD [4], and Megahit [6]. We provide a detailed analysis on how each one behave on metagenome data.

Velvet [3] is a group de Bruin graph-based sequence assembly methods for very short reads that can both remove errors. It also uses read pair information to resolve a large number of repeats. The error correction algorithm merges the sequences that belongs together. Then the repeat solver algorithm separates parts that share overlaps.

Spades [5] is an assembler for both single-cell and standard (multicell) assembly. SPAdes generates single-cell assemblies and provides information about genomes of uncultivatable bacteria that vastly exceeds what may be obtained via traditional metagenomics studies.

IDBA-UD [4] is a de Bruijn graph approach for assembling reads from single cell sequencing or metagenomic sequencing technologies with uneven sequencing depths. IDBA-UD uses multiple depth-relative thresholds to remove erroneous k-mers in both low-depth and high-depth regions. It also uses paired-end information to solve the branch problem of low-depth short repeat regions. It applies and error correction step to correct reads of high-depth regions that can be aligned to high confident contigs.

Megahit [6] is a new approach that constructs a succinct de Bruijn graph using multiple k-mers, and uses a novel ”mercy k-mer” approach that preserves low-abundance regions of reads. It also uses GPUs to accelerate the graph construction.

In this paper, we evaluate the assemblies resulted from the four different assemblers based on several quality metrics and using several preprocessing treatments.

Materials and Methods

Datasets

Podar (write correct name) datasets where downloaded from XX. The dataset represent XX brief description for the data.

Pre-assembly Treatments

We assembled the reads using a combination of different preprocessing and assembly approaches. The preprocessing treatments are:

1. **Quality Filtering:** In this treatment, low quality bases were trimmed and low quality reads were removed using trimmomatic [1]. After quality trimming reads were either directly assembled, or first preprocessed with digital normalization and then assembled. The original datasets contains 5536289548 base pairs and 54814748 sequences in the left pair and 5536289548 base pairs and 54814748 sequences in the right pair.

   After quality filtering, the paired-ended file contains 10547795822 base pairs 104433622 sequences while the single-ended file contains 184437913 base pairs and 1893243 sequences.

2. **Digital Normalization:** Digital normalization works after sequencing data has been generated, progressively removing high-coverage reads from shotgun data sets. This normalizes average coverage to a specified value, reducing sampling variation while removing reads, and also removing the many errors contained within those reads. This data and error reduction results in dramatically decreased computational requirements for de novo assembly. Moreover, unlike experimental normalization where abundance information is removed prior to sequencing, in digital normalization this information can be recovered from the unnormalized reads [2] After digital normalization, the pair ended file contains 1687588894 base pairs and 16853716 sequences while the single ended file contains 5859253 base pairs and 64638 sequences.

3. **Partitioning:** In this treatment, we partitioned the filtered data set based on de Bruijn graph connectivity and assembled each partition independently. Subsequently, partitioning separates reads based on transitive connectivity, resulting in easily assembled subsets of reads.

Metagenomes Assembly

We assembled the reads using four different assemblers; Velvet [3], Idba [4], Spades [5], and Megahit [6] in combination with different preprocessing treatments; quality filtering, digital normalization, and partitioning. We examined the assembly quality of each assembler and treatment sing Quast [7].

Results

Metagenomes Metrics

Table 1 shows various quality metrics for the results of the assembly using combinations of four different assemblers and different preprocessing treatments. The unaligned length is the total length of all unaligned regions in the assembly. The unaligned length for assembly using velvet is 8977149, 10909693, and 11317834 using quality filtered reads, digital normalization, and partitioning respectively. For IDBA assembly, the unaligned length is 10709716, 10637811, and 10644357 using quality filtered reads, digital normalization, and partitioning respectively. For Megahit assembly, the unaligned length is 10686421, 10581435, and10564244 using quality filtered reads, digital normalization, and partitioning respectively. For SPAdes assembly, the unaligned length is 10597529, 10621398, and 10500235 using quality filtered reads, digital normalization, and partitioning respectively.

The genome fraction % is the percentage of aligned bases in the reference. A base in the reference is aligned if there is at least one contig with at least one alignment to this base. For Velvet assembly, the genome fraction percentage is 72.949 %, 89.043%, and 88.879% using quality filtered reads, digital normalization, and partitioning respectively. The genome fraction percentage of IDBA assembly is 90.969 %, 91.003%, and 90.082% using quality filtered reads, digital normalization, and partitioning respectively. For SPAdes assembly, the genome fraction percentage is 90.424%, 90.173%, and 89.272% using quality filtered reads, digital normalization, and partitioning respectively. The genome fraction percentage of megahit assembly is 90.358%, 89.92%, and 88.769% using quality filtered reads, digital normalization, and partitioning respectively.

Misassembled contigs length is the total number of bases in misassembled contigs . For Velvet assembly, misassemble contigs length is 631, 3104, and 3337 using quality filtered reads, digital normalization, and partitioning respectively. For IDBA assembly, misassemble contigs length is 1032, 916, and 828 using quality filtered reads, digital normalization, and partitioning respectively. Misassembled contains length for SPAdes is 752, 881, and 654 using quality filtered reads, digital normalization, and partitioning respectively. For Megahit assembly, misassemble contains length is 648, 780, and 677 using quality filtered reads, digital normalization, and partitioning respectively.

Table 1. Assembly Quality Metrics

Treatment/Quality Metric	Quality Filtering	Digital Normalization	Partition
(1) Velvet			
Genome Fraction	72.949	89.043	88.879
Unaligned Length	8,977,149	10,909,693	11,317,834
Misassembled contigs length	16566891	25594315	16922852
N50	38028	18944	8504
(2) Idba			
Genome Fraction	90.969	91.003	90.082
Unaligned Length	10,709,716	10,637,811	10,644,357
Misassembled contigs length	21777032	27668818	18440791
N50	4,977,3	4,782,8	2,657,5
(3) Spades			
Genome Fraction	90.424	90.173	89.272
Unaligned Length	10,597,529	10,621,398	10,500,235
Misassembled contigs length	28238787	23103154	14338099
N50	4,277,3	3,558,0	2,231,9
(4) Megahit			
Genome Fraction	90.358	89.92	88.769
Unaligned Length	10686421	10581435	10564244
Misassembled contigs length	11927502	17319534	11814070
N50	35254	35427	17492

Time and memory utilizations for assemblies using different treatments

Table 2 shows the running time and memory utilizations for four assemblers and different reads treatments.

For Velvet assemblies, it took $\sim$ 60 hours using quality filtered reads, while it took only $\sim$ 6 hours using digital normalizations and $\sim$ 4 hours using partitioning which is approximately 10% and less of time utilized using quality filtered reads. For IDBA assemblies, it took $\sim$ 33 hours using quality filtered reads, while it took $\sim$ 6 hours using digital normalization and $\sim$ 8 hours using partitioning, approximately less than $\sim$ 7%of time utilized using quality filtered reads. SPAdes assemblies utilized $\sim$ 67 hours using quality filtered reads while it took $\sim$ 15 hours and $\sim$ 7 hours using digital normalization and partitioning respectively less than 5% of time utilized using quality filtered reads.

For Velvet assemblies, it used used $\sim$ 1594851536 KB of memory using quality filtered reads, while it used one $\sim$ 827412304 KB and $\sim$ 1156729920 KB of memory when applying digital normalization and partitioning respectively. For IDBA assemblies, it used used $\sim$ 129853424 KB of memory using quality filtered reads, while it used one $\sim$ 104736448 KB and $\sim$ 93584624 KB of memory when applying digital normalization and partitioning respectively. For SPAdes assemblies, it used used $\sim$ 129853424 KB of memory using quality filtered reads, while it used one $\sim$ 104736448 KB and $\sim$ 93584624 KB of memory when applying digital normalization and partitioning respectively.

Table 2. Running Time and Memory Utilization

Treatment/Quality Metric	**Quality Filtering**	**Digital Normalization**	**Partition**
(1) Velvet			
Running Time	60:42:52	6:48:46	4:30:36
Memory Utilization in KB	1594851536	827412304	1156729920
(2) Idba			
Running Time	33:53:46	6:34:24	8:30:29
‘ **Memory Utilization in KB**	129853424	104736448	93584624
(3) Spades			
Running Time	67:02:16	15:53:10	7:54:26
Memory Utilization in KB	400340512	127423856	129715072
(4) Megahit			
Running Time	1:52:55	0:30:23	1:23:28
Memory Utilization in KB	35034096	19805888	198756832

More about misassembles

S̃till I need an experiment to investigate mis-assemblies more

Mapping assemblies to quality filtered reads

We estimated the percentage of unaligned sequences by each assembly treatment and using the four assemblers. We mapped the quality filtered reads to each assembly. Then we extracted the unaligned sequences to each assembly. Table refreads-mapping shows the percentages of unaligned sequences from quality filtered reads to each assembly treatment using the four assemblers under study. For all treatments assemblies, the full set of trimmed reads were used for mapping. Default parameters were used, and both paired ends and singletons were mapped. Samtools [8] was used for format conversion from SAM to BAM format, and also to calculate the percentage of mapped reads.

Mapping unaligned reads of all assemblers and treatments to the unaligned reads of IDBA assembly using quality treatment

In this experiment, we mapped unaligned reads of each assembly with different treatments to the the unaligned reads of idba assembly using quality filtered treatment. The purpose of this experiment is to identify whether the unaligned reads are common.

Table 3. Reads Mapping

Treatment/Quality Metric	Quality Filtering	Digital Normalization	Partition
(1) Velvet			
No. of Unaligned Sequences	8324608	2205698	2697788
(2) Idba			
No. of Unaligned Sequences	495570	549791	1302356
(3) Spades			
No. of Unaligned Sequences	714474	842268	1408063
(4) Megahit			
No. of Unaligned Sequences	467660	622684	1487942

Table 4. Mapping unaligned reads to Idba quality-filtered assembly

Treatment/Quality Metric	Quality Filtering	Digital Normalization	Partition
(1) Velvet			
Genome Fraction	80.613	92.034	98.013
Unaligned Length	2475529	3192491	64539560
(2) Idba			
Genome Fraction	-	91.53	94.738
Unaligned Length	-	498299	37437754
(3) Spades			
Genome Fraction	91.922	93.959	94.826
Unaligned Length	2174574	1951911	2398664
(4) Megahit			
Genome Fraction			
Unaligned Length			

Discussion

Assembly works pretty well

Except for Velvet assembly using quality filtered reads, the genome fraction percentage is 88% or higher. Unaligned length is less than 1% for all assemblers and using different treatments. Misassembled length is less than 1.3% for all assemblers and using different treatments.

Digital normalization and partitioning significantly reduce running time and memory utilizations

The difference between genome fraction percentage using quality filtered reads vs digital normalizations and partitioning doesn't exceed 1%. However, the time and memory resource are reduced a lot using digital normalization and partitioning.

Unaligned reads are common among different assemblers

Mapping the unaligned reads of different assemblies and different treatments to the unaligned reads of IDBA assembly using quality filtered , shows genome fraction percentage is 91% or higher. This means the unaligned reads are common among assemblers and they are likely to be because of contamination.

Acknowledgments

References

References

1. Bolger AM, Lohse M, Usadel B (2014) Trimmomatic: A flexible trimmer for illumina sequence data. Bioinformatics 30: 2114-2120.

2. Brown CT, Howe A, Zhang Q, Pyrkosz AB, Brom TH (2012) A reference-free algorithm for computational normalization of shotgun sequencing data. arXiv e-print 12034802 .

3. Zerbino DR, Birney E (2008) Velvet: algorithms for de novo short read assembly using de bruijn graphs. Genome Research 18: 821-829.

4. Peng Y, Leung HC, Yiu S, Chin FY (2012) Idba-ud: a de novo assembler for single-cell and metagenomic sequencing data with highly uneven depth. Bioinformatics 28: 1420-1428.

5. Bankevich A, Nurk S, Antipov D, Gurevich AA, Dvorkin M, et al. (2012) Spades: A new genome assembly algorithm and its applications to single-cell sequencing. Journal of Computational Biology 19: 455-477.

6. Li D, Liu CM, Luo R, Sadakane K, Lam TW (2014) Megahit: An ultra-fast single-node solution for large and complex metagenomics assembly via succinct de bruijn graph. Bioinformatics .

7. Alexey Gurevich NV Vladislav Saveliev, Tesler G (2013) Quast: quality assessment tool for genome assemblies. Bioinformatics 28: 1072-1075.

8. H L, B H, A W, T F, J R, et al. (2009) The sequence alignment/map format and samtools. Bioinformatics 25.

Figure Legends

Tables

Supporting Information Legends