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

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 [**?**]. 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 [**?**] 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 [**?**], Idba [**?**], Spades [**?**], and Megahit [**?**] in combination with different preprocessing treatments.

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. Table 3 shows the percentage of unaligned sequences when mapping the raw reads to the results assembly.

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 Idba, digital normalization reduces 27 hours in the running time. While partitioning reduces 25 hours in the running time. For SPAdes, digital normalization reduces 52 hours in the running time. While partitioning reduces 59 hours in the running time. For Velvet, digital normalization reduces 54

hours in the running time. While partitioning reduces 56 hours in the running time. For Megahit, digital normalization reduces XX hours in the running time. While partitioning reduces XX hours in the running time.

Digital normalization and Partitioning also reduce memory requirements. For Idba, digital normalization reduces XX KB of memory utilization. While partitioning reduces XX KB of memory utilization. For SPAdes, digital normalization reduces XX KB of memory utilization. While partitioning reduces XX KB of memory utilization. For megahit, digital normalization reduces XX KB of memory utilization. While partitioning reduces XX KB of memory utilization.

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 [**?**] 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

Digital normalization and partitioning significantly reduce running time and memory utilizations

Or Megahit Idba shows the lowest number of unaligned reads among different treatments

Velvet shows the highest number of unaligned reads among different treatments

1. **How well do different assemblers recover metagenomes?**
2. **Why do contigs misassemble/what is in common between mis-assemblies**
3. **How do different treatments affect assembly quality?**

4. **How do the different treatments affect computational requirements?**
5. **—stuff from results –may be combine results and discussion**
6. **Assembly works pretty good**
7. **Discuss all/most of the metrics and justify them – not only pick one or 2 important metrics**
8. **talk about misassembles and justify them**
9. **talk about unalignment and justify them**
10. **explain how every treatment – leads to memory or time enhancements**

Acknowledgments

References

@articleBrown2012, Author = C Titus Brown and Adina Howe and Qingpeng Zhang and Alexis B Pyrkosz and Timothy H Brom, Journal = , Month = May, Number = , Title = A reference-free algorithm for computational normalization of shotgun sequencing data, Volume = , Year = 2012

Figure Legends

Tables

Supporting Information Legends