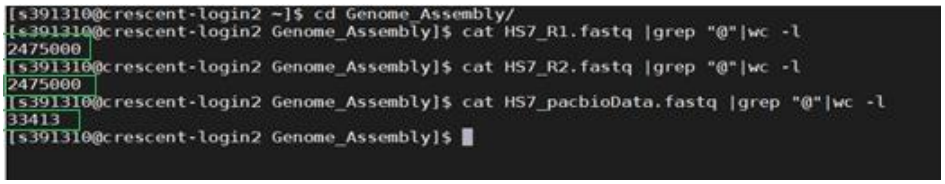
Genome Assembly Report

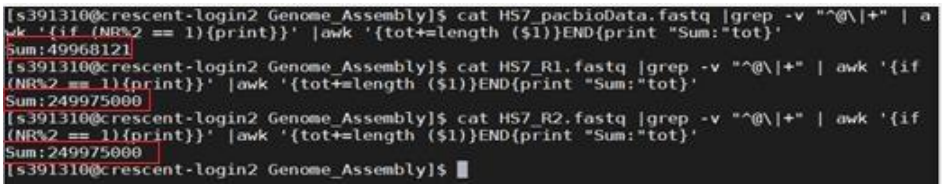
1. Introduction

   In this advanced sequencing informatics assignment, three different de-novo genome assemblies were tried namely SOAPdenovo2 (short read assembly) and two hybrid assemblies (MaSuRCA and HybridSPades). Along with which the genome size and the quantitative statistical metrics was calculated.

2. Quality control

   a. The files were copied from the original folder and then the raw reads were unzipped using "gunzip" with two short reads (illumina) and one long read (pacbio). And the number of reads with number of bases were found out to get a rough size of the dataset.

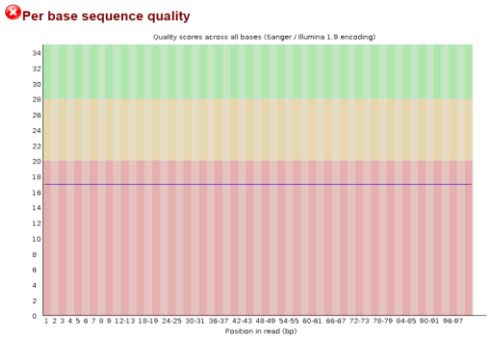




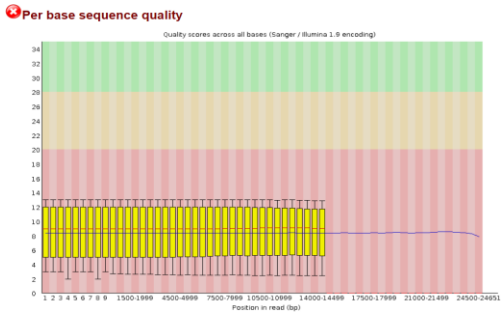
   We were given illumina PE reads of 101 x2 with insert size 350bps +/- 50, which is basically the no. of bases/no. of reads;= 249975000/2475000=~101 similarly the length of long reads of pacbio could be found out by;=49968121/33413~ 1495.5~1495. After which the quality of data was determined using "fastQC".

   b. The data quality was quite below the average with a phred score of 17 for the short reads and ~ 9 for the long reads. Also the sequence length distribution was quite large initially for the long reads. So the error correction was carried out by using "Corrector_HA" before which "KmerFreq_HA" for the k-mer spectrum was done for the short reads.

      - The parameters were kmer size as k=15 (due to max size of computer memory) and the threshold quality score of 33. Which resulted in creating the corrected files for the "R1" and "R2" reads which was used for the assemblers.



(for illumina reads)



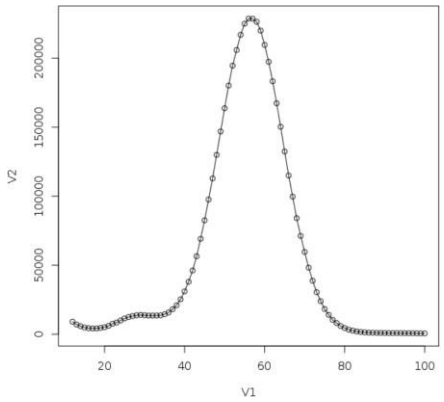
(pac bio reads)

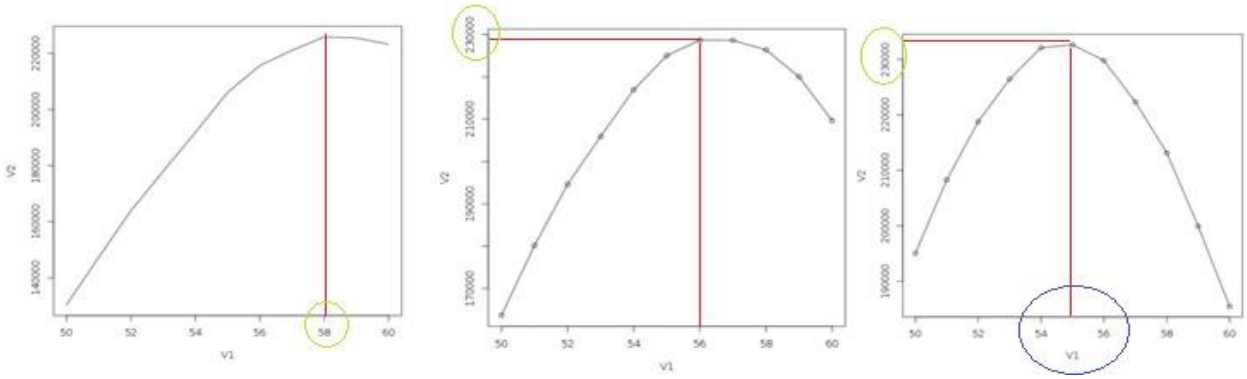
3. K-mer Analysis

a. The software “jellyfish” was used to find out the k-mer count and then in R histogram was plotted for the same. Both the corrected short reads were given as input.

b. The parameters selected for the kmer analysis were:-
   - kmer size (-m) was tested from 15,17,19,21,23,25,27 and 29. (odd numbers are usually taken as they have central nucleotide).
   - Number of threads (-t) was taken as 2.
   - Memory size allocation (-G) was taken as 1Gb.
   - And for cannonical allocation, (-C) was given to it.

c. Histograms were plotted for the outputs from the above parameters. And the top three kmer values based on the histograms after removing the errors (initial few points) were found out to be:- 25,27 and 29 respectively as shown in the figure below.





(25) (27) (29)

Out of which k-mer value for the genome size estimation was selected to be large enough to capture the complexity of the data and not so large that k-mer becomes too rare, so the x axis value at -m =29 was taken as it corresponds to the highest peak (~230000) with the the number of 29mers being as 55 !
So for all the three assemblers the k-mer size was taken as 55.

4. Genome Size estimation

- The number of total Kmers were calculated after removing the first 12 points corresponding to x axis (as they computed for the error) And Genome size was calculated by dividing the totKmer to the position at peak (ie 55).

```
> totKmers
[1] 269190650
> G=totKmers/55
> G
[1] 4894375
```

So the genome size was found out to be close to 4900000 (~4.9 Mb).

GENOME ASSEMBLIES

1) SOAPdenovo2

- The first assembly that was tried was based on short reads;SOAPdenoovo2.For this assembly there were two options, namely; SOAPdenovo-63mer and SOAPdenovo-127 mer. Out of which SOAPdenovo-63 mer was selected as 63 Gb of RAM would be sufficient for this assembly and the computational resources would be optimum for every user on the HPC.
- The corrected files “HS7_R1.fastq.cor.pair_1.fq” and “HS7_R2.fastq.cor.pair_2.fq” was used in the nano file that was created and named as “pek55” with the parameter as:
  a) Avg_ins= 350, which is essentially the average insert size of the given short read paired end illumina sequence(as provided in the assignment)
  b) reverse_seq=0, as we have paired end reads, which basically results in two separate sequencing reads for each fragment. For the reverse orientation only , this value is taken as 1.
  c) Asm_flags=0 was provided, as the input file given was already corrected using the “Corrector_HA” and this option with the value “0” will perform assembly without any error correction and will use comparatively less computational resources.
  d) rank=1 basically gives the priority to the job submitted , the value “1” gives it the highest priority.
  e) The q1 and q2 options provide the paths for the two corrected paired end reads.
- In the next step the De-Brujin graph was compacted with the help of “sparse_graph” command and the following parameters were provided:
  a) SOAPdenovo-63mer is basically calling the assembler of ram usage “63Gb” for our task
  b) sparse_pregraph allows us to save computational power by using a simplified graph that will only retain the most informative k-mers and their corresponding links to other k-mers.
  c) -s is the input file name (“soapPE.conf”)

d) -K is the the size of the k-mer which was selected to be “55” as found out from the k-mer analysis section above.
e) -z is actually the size of the hash table which is created for k-mer counting and can be closely estimated by the genome size which was found out to be approx. 4.9 millions (from the above section) so this is the value which was provided to this parameter. (“4900000”)
f) -p is the number of threads that is to be used by the parallel processing which was set to “2”
g) -o is the name of the output file which was called as “peK55.

- After creating sparse pregraph, the next step was to create contigs from this sparse pregraph. And the parameters decided were:
  a) contig, which will output the contigs file
  b) -g and -s are were the output prefix and the location of thfile “soapPE.conf”
  c) -p 2 specifies 2 threds to be used in the computer resource.
- After the contigs creation, the quality of the assembled sequence was found out by the tool gnx.jar which provided stats like :-

```
Singularity gamod.simg:~/Genome_Assembly> java -jar /usr/local/bin/gnx.jar peK55.contig
Results for peK55.contig
Total number of sequences:          3585
Total length of sequences:          5156763 bp
Shortest sequence length :          56 bp
Longest sequence length  :          77632 bp
Total number of Ns in sequences:    0
N50:    15018   (101 sequences) (2587342 bp combined)
```

- The most important parameter is the N50. Which means that the half of the total assembly length is contained within sequences that are at least 15,018 base pair long and also the there are 101 sequences in the assembly that contribute to this N50 and the combined length of all the sequences in the assembly is 2587342 base pairs (2.5M).
- Apart from this parameter the other three most important parameters are :-
  a) the total number of sequences which should be lesser as it tells that every contig has more number of bases which is a good indication.
  b) Longest sequence and shortest sequence length, as when the N50 is arranged its usually from the largest to smallest

sequence length. And in the assembly the larger contig corresponds to a better insight into the genomic structure and is easier for annotations. While the shortest sequence and its numbers tells that there are more fragmentations in the final genome assembly. A high skewness towards short reads indicate a more difficulty in accurate mapping. And the longest sequence was 77632 bp while the shortest was 56 bp long

c) Number of gaps should be lesser as it tells us that the genome assembly is more continuous and in this assembler the number of N's reported were 0 which is an indication of a good quality genome assembly.
Note:- As there were no jump_files , scaffolding was not possible with this assembler.

2) MaSuRCA

- The second assembly that was tried was masurca which is a hybrid assembly taking short reads as well as the long reads.
- First of all the masurca_config file was created by running the masurca genome assembly in which there were some parameters which were modified as follows:-
  1) PE which is basically the paired end reads and it was changed to 350 (as per given insert size) with an error of 50. And the paths for the short reads were given.
  2) As jump files were not there the parameter "JUMP" was commented
  3) The path for the long read (pacbio) was provided.
  4) The GRAPH_KMER_SIZE was set to 55 which is the k-mer size (as mentioned in k-mer analysis)
  5) USE_LINKING_MATES was set to 0 which means that assembly will also use long reads and not just the short reads
  6) USE_GRID was set to 0 as corrected files were only provided.
  7) LHE_COVERAGE was given as 30 which is the coverage by the longest read to use.
  8) KMER_COUNT_THRESHOLD was set to 1 to use all the k-mers for error correction.

9) CLOSE_GAPS was set to 1 to close the gaps in scaffolds with the illumina data.
10) JF_SIZE was the default value of 2000 M (hash size of the jellyfish, which is too large and is just as precaution)
11) SOAP_ASSEMBLY was set to 0 to tell the program to not use SOAPdenovo module.

- The file was run after assembling and the following result was obtained from the genome.scf.fasta after using the gnx tool:-

```
Results for CA.mr.41.15.17.0.029/final.genome.scf.fasta
Total number of sequences:          17
Total length of sequences:          4973017 bp
Shortest sequence length :          12850 bp
Longest sequence length  :          1319902 bp
Total number of Ns in sequences:    0
N50:    517108  (3 sequences)   (2497098 bp combined)
```

- The N50 parameter indicates that a total of 2.4 M base pairs were combined from the three sequences which is quite close to the previous assembly (SOAPdenovo2). While the third longest sequence in the assembly is 517108 and it is the point at which 50% of the total assembly length is contained in sequences of that length or longer.
- Other important results from this statistics is:-
  a) the genome assembly was assembled to only 17 sequence scaffolds which is quite good as a lesser number sequences indicates that it is more contiguous and has fewer misassembles/redundant sequences.
  b) total length of the sequence is quite close to the estimated genome size ~4.9 Mb which shows that the value of kmer size as 55 was better.
  c) The shortest sequence length is 12850 bp which is better than the shortest sequence of soapdenovo2 (56) which shows that the fragmentation and the number of fragmentation is also less.(as total no of seq is 17).
  d) The longest scaffold that was obtained was quite huge with approx. 1.3 Mbp .

3) HybridSPAdes

- The third assembly done was again a hybrid one; spades .
- As spades is a resource hungry assembler ,it had to be submitted through a .sub file . And the parameters for it were:-
  a) -o which is the name of the output directory for the assembly results.(spadeu)
  b) –s1 and –s2 are the paths of both the short illumina reads
  c) –pacbio is the path to long read (pacbiodata)
  d) -t is the number of threads which was taken as 4
  e) -m is the memory allocation for assembly which was 4 GB
  f) –kmers was not specified because when it is not provided, by default it will select the best k-mer length.

- Spades creates multiple k-mer lengths to construct the de Brujin graphs which are then used to identify the overlaps between reads and to assemble contigs. And the reason the kmer was not selected was to see the top three results that the software selects. And if Kmer 55 would not have been there, manually placing the kmer value as 55 would be the go to choice so that all the three assemblers are comparable. And the top three Kmer values which the spade software selected was;21,33 and 55 !

```
Results for scaffolds.fasta
Total number of sequences:          286
Total length of sequences:          4959701 bp
Shortest sequence length :          56 bp
Longest sequence length  :          701839 bp
Total number of Ns in sequences:    17151
N50:    309122  (6 sequences)   (2672102 bp combined)

Singularity gamod.simg:~/Genome_Assembly/spades/spadeu> S
```

(result for the kmer55)

From the spade results we can see that the N50 base pairs combined is the maximum uptill now ,but that is not coclusive enough to regard it as the best assembler as it has a lot of N's ~17000 which are a lot of gaps. Although the spade assembler has comparatively very less number of sequences than the Soapdenovo2.

Conclusion

Comparing the N50 values of all the three assemblers, spades performed the best with 2672102 bp combined while the third one was masurca with 2497098 bp combined which if we see is not much of the difference between the two in scale of the dna size we are dealing with. Although this is not the absolute measure ,if we compare the number of sequences produced by all the three, masurca was the best with only 17 ! which is very less than the 3583(soap) and 286(spades) meaning we managed to get more contiguous sequence with masurca. And because of this the shortest sequence in masurca was also longer than the shortest sequence of all the assemblers. If we compare the total length of the sequences, all the three were nearly same with around 5Mbp which is also quite close to the estimated genome size (4.9Mbp) . The number of N's (gaps) is another important factor to be considered, wherein spades had 17000 N's while for the other assemblers it was 0.

Also choosing kmer size as 55 was a good choice as confirmed by the spades assembler (running it without the Kmer size gave top three kmer values in which 55 was also there)

So overall Masurca performed the best as it had least number of sequnces, no gaps and the N50 value was quite close to the other two assemblers.