BIOL550-Bioinformatics- Group Project

F17_T3

Dawei Wang (iits_18)

Jiaxing Miao (iits_14)

Zonglin Yang (iit_21)

1. Preparation

   (1) Decompress illumina data (R1.fastq.gz & R2.fastq.gz)

   gzip -d --force Illumina_R1.fastq.gz

   gzip -d --force Illumina_R2.fastq.gz

   (2) Decompress Nanopore data and transfer it into fastq format

   tar -zxvf nanopore.tar.gz

   cd FAST5

   poretools stats --type 2D *.fast5

   poretools fastq --type 2D *.fast5 > Nanopore.2D.fastq

Therefore we have R1.fastq & R2.fastq & Nanopore.2D.fastq and put them into different files: illumina data into [Illumina] file and nanopore data into [Nanopore] file so that bring convenience to further operation.

2. Examining quality scores

   The quality scores of all 3 data are appropriate.

   For nanopore data, the quality score is between 8-14, so the data is well fitted. We keep position 1-9 in order for maintaining the integrity of data.

   Besides, there are no adapters in those data.

3. Assembling by Ray, Canu and SPAdes

   Make two directories for further operation. And copy corresponding files into right directories.

   Use Kmergenie to help choose best kmer.

   For Illumina

   ls *.fastq > list

   kmergenie list

   (1) Using Ray assemble illumina data

   nano Ray.sh

   [in bash script]

   #!/usr/bin/bash

echo "Running Ray(k61)";
mpirun -np 10 Ray -k61 -p Illumina_R1.fastq Illumina_R2.fastq -o IllRay-k61;
echo "Ray(k61) Done";
echo "Running Ray(k81)";
mpirun -np 10 Ray -k81 -p Illumina_R1.fastq Illumina_R2.fastq -o IllRay-k81;
echo "Ray(k81) Done";
echo "Running Ray(k101)";
mpirun -np 10 Ray -k101 -p Illumina_R1.fastq Illumina_R2.fastq -o IllRay-k101;
echo '3kmers Ray is done.'
we choose 3 different Kmers and use screen and bash script to finish the process.

(2) Using Canu assemble nanopore data

canu -p MinCanu -d MinCanu -maxThreads=10 -genomeSize=2.19m -nanopore-raw *.fastq

canu -p OrCanu -d Orcanu -maxThreads=10 -genomeSize=2.6m -nanopore-raw *.fastq

we check on NCBI, the interval of genome size for this bacteria is (2.18819, 2.71143), so we choose a lower size(2.19m) and median size(2.60m).

(3) Using SPAdes assemble illumina & nanopore data

spades.py --careful --nanopore Nanopore.2D.fastq --pe1-1 Illumina_R1.fastq --pe1-2 Illumina_R2.fastq -o SPAdes

Rename and copy all Contig files to /Contigs.

4. Blast: blastn and faSomeRecords by perl: Operate under /Contigs

(1) megablastn: using bash script; using screen to operate all blastn analysis

nano ~/.bash_profile

[add this line below in]

export BLASTDB=/media/Data_1/NCBI/NT

source ~/.bash_profile

nano blastn_Ray.sh # running Ray under Kmer=61, 81, 101

[in bash script]

```
#!/usr/bin/bash
echo "Running blastn with Ray_k61";
blastn -task megablast -query Ray_k61_Contigs.fasta -db mini_nt -out Ray_k61_Contigs.blastn -evalue 1e-10 -culling_limit 1 -outfmt '6 qseqid sseqid bitscore evalue staxids sskingdoms sscinames' -num_threads 20;
echo "Done";
echo "Running blastn with Ray_k81";
blastn -task megablast -query Ray_k81_Contigs.fasta -db mini_nt -out Ray_k81_Contigs.blastn -evalue 1e-10 -culling_limit 1 -outfmt '6 qseqid sseqid bitscore evalue staxids sskingdoms sscinames' -num_threads 20;
echo "Done";
echo "Running blastn with Ray_k101";
blastn -task megablast -query Ray_k101_Contigs.fasta -db mini_nt -out Ray_k101_Contigs.blastn -evalue 1e-10 -culling_limit 1 -outfmt '6 qseqid sseqid bitscore evalue staxids sskingdoms sscinames' -num_threads 20;
blastn -task megablast -query Canu_Contigs.fasta -db mini_nt -out Canu_Contigs.blastn -evalue 1e-10 -culling_limit 1 -outfmt '6 qseqid sseqid bitscore evalue staxids sskingdoms sscinames' -num_threads 20
echo 'All Ray blastn Done';
```

For Canu:

```
blastn -task megablast -query Canu_Contigs.fasta -db mini_nt -out Canu_Contigs.blastn -evalue 1e-10 -culling_limit 1 -outfmt '6 qseqid sseqid bitscore evalue staxids sskingdoms sscinames' -num_threads 20
blastn -task megablast -query 02_Canu_Contigs.fasta -db mini_nt -out 02_Canu_Contigs.blastn -evalue 1e-10 -culling_limit 1 -outfmt '6 qseqid sseqid bitscore evalue staxids sskingdoms sscinames' -num_threads 20
```

For SPAdes:

```
blastn -task megablast -query SPAdes _Contigs.fasta -db mini_nt -out SPAdes
```

_Contigs.blastn -evalue 1e-10 -culling_limit 1 -outfmt '6 qseqid sseqid bitscore evalue staxids sskingdoms sscinames' -num_threads 20

(2) Perl Script and faSomeRecords

nano v.pl

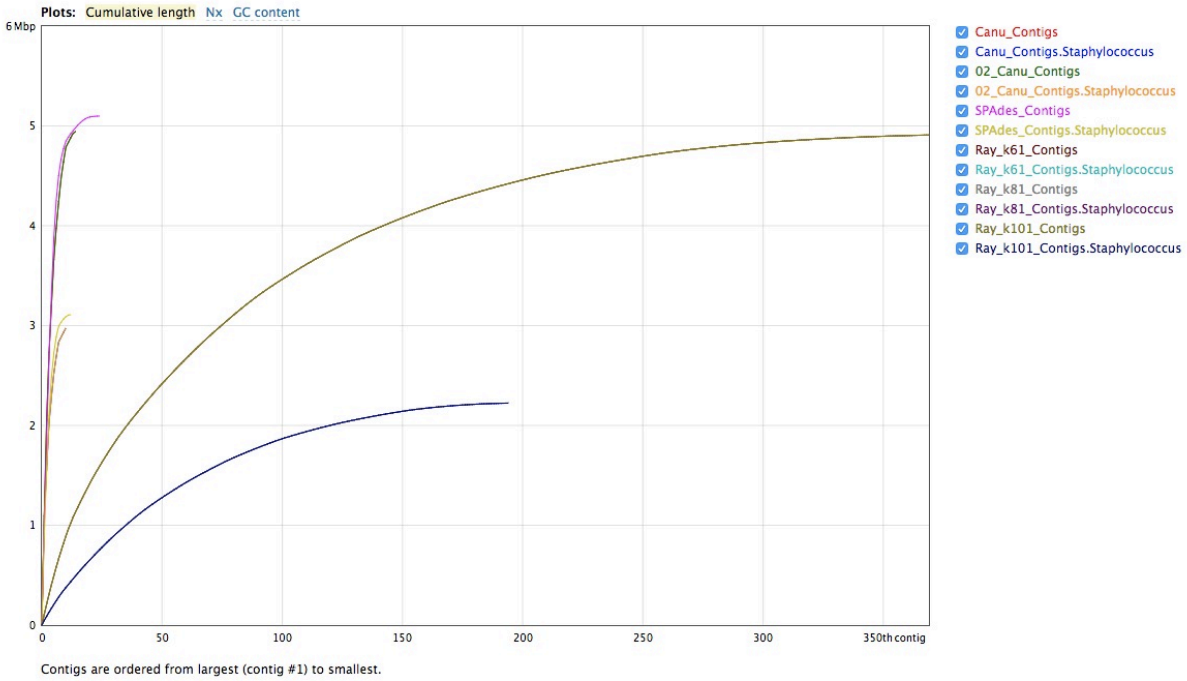
[in perl script]

```
#!/usr/bin/perl
$usage = "USAGE -> perl v.pl *.blastn";
die "\n$usage\n" unless @ARGV; #usage introfuction

while ($file = shift @ARGV){
        open IN, "<$file"; #open the .blastn file
        $file =~ s/.blastn$//;
        open OUTS, ">$file.Staphylococcus_list";
        open OUTC, ">$file.Contaminant_list";
        while ($line = <IN>){
                chomp $line;
                if ($line =~ m/.+Staphylococcus/){
                        @contig = split("\t",$line);
                        print OUTS "$contig[0]\n";
                        }
                else  {
                        print OUTC "$contig[0]\n";
                        }
        }
     $input = "$file.fasta"; #for system usage
        $output = "$file.Staphylococcus.fasta";
        $list = "$file.Staphylococcus_list";
        $listC = "$file.Contaminant_list";
        $outputC = "$file.Contaminant.fasta";
```

```
        system "faSomeRecords $input $list $output";
        system "faSomeRecords $input $listC $outputC";
    }
    system 'echo "Job is done, ALL multifasta file has been created."';
```

5. Quast & Analysis

quast.py -o QuastResult Canu_Contigs.fasta Canu_Contigs.Staphylococcus.fasta 02_Canu_Contigs.fasta 02_Canu_Contigs.Staphylococcus.fasta SPAdes_Contigs.fasta SPAdes_Contigs.Staphylococcus.fasta Ray_k61_Contigs.fasta Ray_k61_Contigs.Staphylococcus.fasta Ray_k81_Contigs.fasta Ray_k81_Contigs.Staphylococcus.fasta Ray_k101_Contigs.fasta Ray_k101_Contigs.Staphylococcus.fasta



The Canu results are different in genome size, but their graphs are coincided.

For SPAdes, the wathet blue line is the best due to faster reaching to maximize so that the number of contig is smaller than others.

For Ray, because of the very negligible difference in Kmer=61, 81 and 101, we decide to use the Kmer=81.

Among these three different approaches, we believe that SPAdes performs better, the hybrid data would provide more reliability. Besides that, SPAdes has a higher N50 than others, as well as the largest length and assemble accuracy.

Statistics without reference	Ray_k61_Contigs	Ray_k81_Contigs	Ray_k101_Contigs	Ray_k81_Contigs.Staphylococcus
# contigs	369	369	369	194
Largest contig	106 500	106 500	106 500	46 394
Total length	4 908 972	4 909 299	4 909 182	2 223 849
N50	25 750	25 750	25 750	19 073

Statistics without reference	Canu_Contigs	02_Canu_Contigs	Canu_Contigs.Staphylococcus
# contigs	14	14	10
Largest contig	1 093 166	1 093 166	1 093 166
Total length	4 943 626	4 943 626	2 974 536
N50	666 714	666 714	470 912

Statistics without reference	SPAdes_Contigs	SPAdes_Contigs.Staphylococcus
# contigs	24	12
Largest contig	983 817	888 174
Total length	5 098 693	3 110 355
N50	794 189	553 249

6. PROKKA Annotation & Artemis & Number of Proteins

(1) PROKKA Annotation: We annotate 3 assembly: Canu, Ray(Kmer=81) and SPAdes

prokka *.Staphylococcus.fasta --outdir prokka_annotation_* --prefix BIO550_ --metagenome --kingdom Bacteria --increment 10 --compliant --addgenes --mincontiglen 200 --centre –protein

generating 3 folders containing 11 files, individually. The *.gbk(genebank) file is used for splitting and annotation. The *.faa file is used for calculation of number of protein.

(2) Artemis

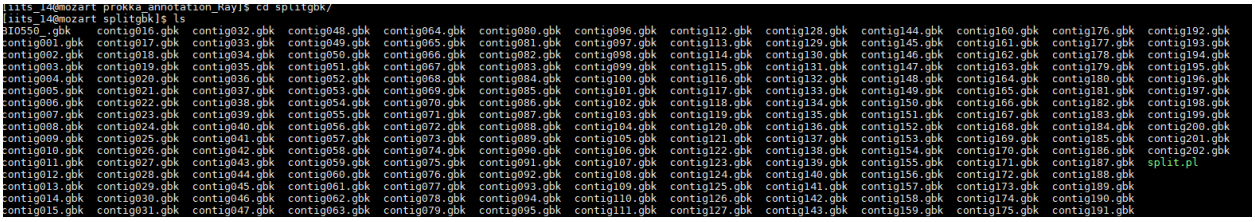
nano split.pl

[in perl script]

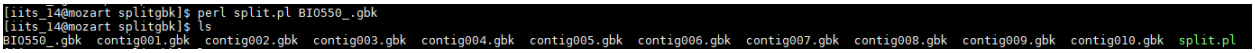
```
#!/usr/bin/perl

my $file = shift (@ARGV);
```

```
open IN, "<$file";
my $output;
while (my $line = <IN>){
    chomp $line;
     if ($line =~ /^LOCUS\s+(\S+)/){
                $output = "$1.gbk";
                open OUT, ">>$output";
                print OUT "$line\n";
                close OUT;
     }
     else{
                open OUT, ">>$output";
                print OUT "$line\n";
                close OUT;
     }
}
./split2.pl BIO550_.gbk ##running the perl
```

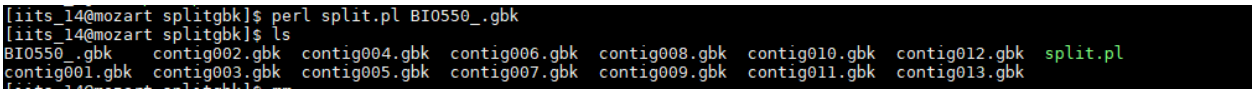
For Ray



For Canu



For SPAdes



(3) Number of Proteins: under each folders, using command line below

grep -c '>' BIO550_.faa

Approach	Ray	Canu	SPAdes

Protein Number	2109	6068	2891

Analysis: According to NCBI database, the interval of genome size is (2.18819, 2.71143). The rough calculation gives the number of protein equaled to 2400. Ray and SPAdes analysis are very approximately to it, but Canu analysis is very huge. The reason we assume is Ray is just reflect the approximate protein number, SPAdes is a ref based on Ray. Canu, on the other hand, would be not that precise, protein number often triple than regular number.