

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 -evaluate 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 -evaluate 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 -evaluate 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 -evaluate 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 -evaluate 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 -evaluate 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 sskindoms 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."';

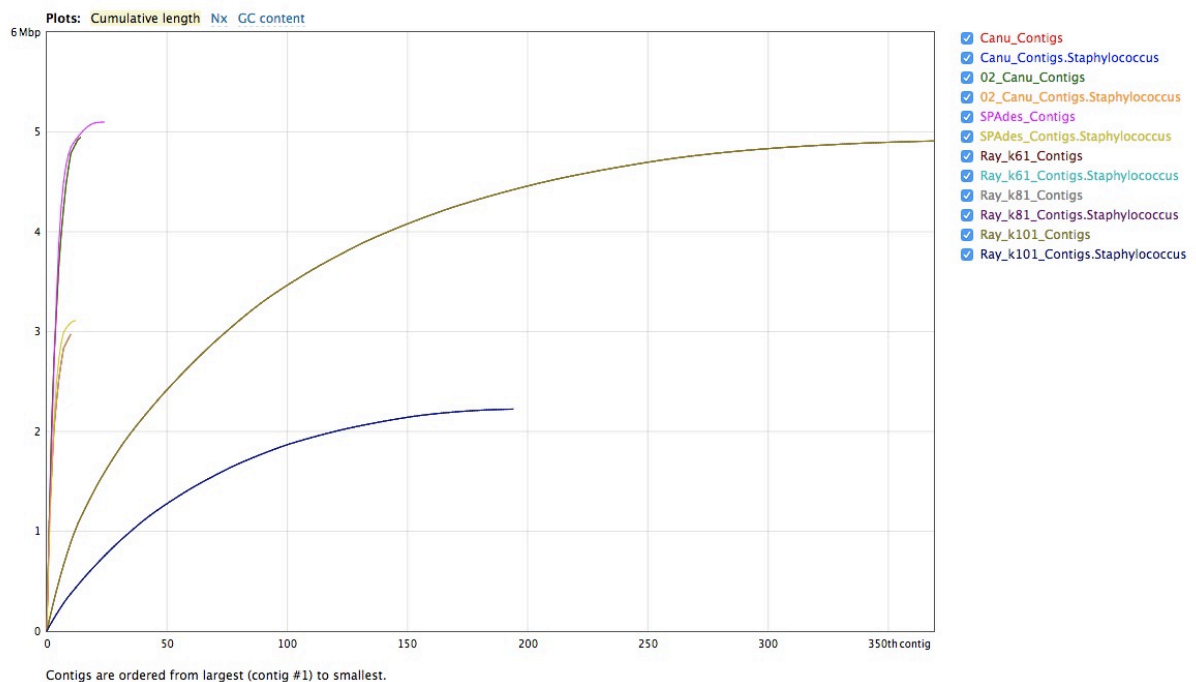
```

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

```

[iits_14@mozart prokka_annotation_Ray]$ cd splitgbk/
[iits_14@mozart splitgbk]$ ls
BIO550_.gbk      contig016.gbk  contig032.gbk  contig048.gbk  contig064.gbk  contig080.gbk  contig096.gbk  contig112.gbk  contig128.gbk  contig144.gbk  contig160.gbk  contig176.gbk  contig192.gbk
contig001.gbk  contig017.gbk  contig033.gbk  contig049.gbk  contig065.gbk  contig081.gbk  contig097.gbk  contig113.gbk  contig129.gbk  contig145.gbk  contig161.gbk  contig177.gbk  contig193.gbk
contig002.gbk  contig018.gbk  contig034.gbk  contig050.gbk  contig066.gbk  contig082.gbk  contig098.gbk  contig114.gbk  contig130.gbk  contig146.gbk  contig162.gbk  contig178.gbk  contig194.gbk
contig003.gbk  contig019.gbk  contig035.gbk  contig051.gbk  contig067.gbk  contig083.gbk  contig099.gbk  contig115.gbk  contig131.gbk  contig147.gbk  contig163.gbk  contig179.gbk  contig195.gbk
contig004.gbk  contig020.gbk  contig036.gbk  contig052.gbk  contig068.gbk  contig084.gbk  contig100.gbk  contig116.gbk  contig132.gbk  contig148.gbk  contig164.gbk  contig180.gbk  contig196.gbk
contig005.gbk  contig021.gbk  contig037.gbk  contig053.gbk  contig069.gbk  contig085.gbk  contig101.gbk  contig117.gbk  contig133.gbk  contig149.gbk  contig165.gbk  contig181.gbk  contig197.gbk
contig006.gbk  contig022.gbk  contig038.gbk  contig054.gbk  contig070.gbk  contig086.gbk  contig102.gbk  contig118.gbk  contig134.gbk  contig150.gbk  contig166.gbk  contig182.gbk  contig198.gbk
contig007.gbk  contig023.gbk  contig039.gbk  contig055.gbk  contig071.gbk  contig087.gbk  contig103.gbk  contig119.gbk  contig135.gbk  contig151.gbk  contig167.gbk  contig183.gbk  contig199.gbk
contig008.gbk  contig024.gbk  contig040.gbk  contig056.gbk  contig072.gbk  contig088.gbk  contig104.gbk  contig120.gbk  contig136.gbk  contig152.gbk  contig168.gbk  contig184.gbk  contig200.gbk
contig009.gbk  contig025.gbk  contig041.gbk  contig057.gbk  contig073.gbk  contig089.gbk  contig105.gbk  contig121.gbk  contig137.gbk  contig153.gbk  contig169.gbk  contig185.gbk  contig201.gbk
contig010.gbk  contig026.gbk  contig042.gbk  contig058.gbk  contig074.gbk  contig090.gbk  contig106.gbk  contig122.gbk  contig138.gbk  contig154.gbk  contig170.gbk  contig186.gbk  contig202.gbk
contig011.gbk  contig027.gbk  contig043.gbk  contig059.gbk  contig075.gbk  contig091.gbk  contig107.gbk  contig123.gbk  contig139.gbk  contig155.gbk  contig171.gbk  contig187.gbk  split.pl
contig012.gbk  contig028.gbk  contig044.gbk  contig060.gbk  contig076.gbk  contig092.gbk  contig108.gbk  contig124.gbk  contig140.gbk  contig156.gbk  contig172.gbk  contig188.gbk
contig013.gbk  contig029.gbk  contig045.gbk  contig061.gbk  contig077.gbk  contig093.gbk  contig109.gbk  contig125.gbk  contig141.gbk  contig157.gbk  contig173.gbk  contig189.gbk
contig014.gbk  contig030.gbk  contig046.gbk  contig062.gbk  contig078.gbk  contig094.gbk  contig110.gbk  contig126.gbk  contig142.gbk  contig158.gbk  contig174.gbk  contig190.gbk
contig015.gbk  contig031.gbk  contig047.gbk  contig063.gbk  contig079.gbk  contig095.gbk  contig111.gbk  contig127.gbk  contig143.gbk  contig159.gbk  contig175.gbk  contig191.gbk

```

For Canu

```

[iits_14@mozart splitgbk]$ perl split.pl BIO550_.gbk
[iits_14@mozart splitgbk]$ ls
BIO550_.gbk  contig001.gbk  contig002.gbk  contig003.gbk  contig004.gbk  contig005.gbk  contig006.gbk  contig007.gbk  contig008.gbk  contig009.gbk  contig010.gbk  split.pl

```

For SPAdes

```

[iits_14@mozart splitgbk]$ perl split.pl BIO550_.gbk
[iits_14@mozart splitgbk]$ ls
BIO550_.gbk  contig002.gbk  contig004.gbk  contig006.gbk  contig008.gbk  contig010.gbk  contig012.gbk  split.pl
contig001.gbk  contig003.gbk  contig005.gbk  contig007.gbk  contig009.gbk  contig011.gbk  contig013.gbk

```

(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.