Progress Update

Farhan Tahir

12/05/2016

Contents

Program's Accuracy

- This topic will describe the following:

1. Preparation of file sample.
2. Output's accuracy result.

Preparation of File Sample

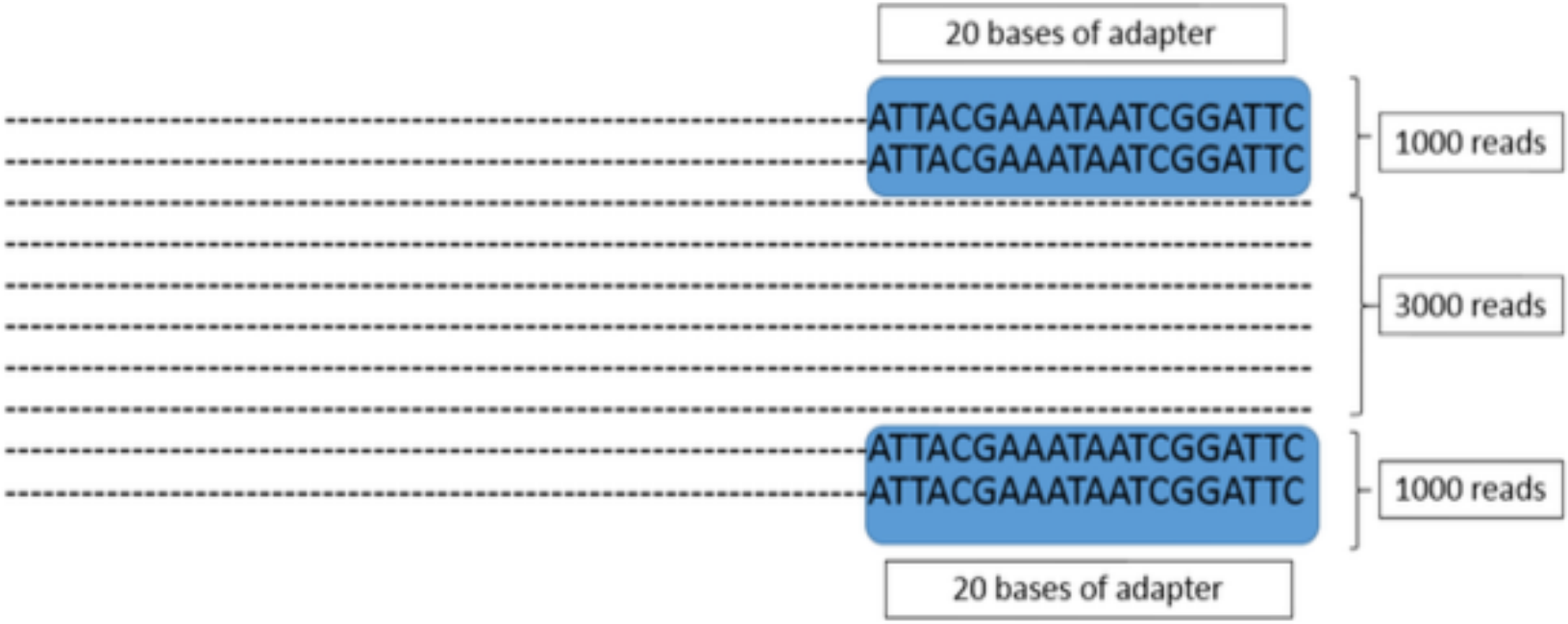
1)Sample file are prepared as FASTQ format.

2)In each sample file, there are 5000 read of sequence, (120 bases/read of sequence):

```
+ first 1000 sequence          : sequence + adapter sequence
                                 adapter sequence)
+ the following 3000 sequence : sequence WITHOUT adapter se
+ last 1000 sequence           : sequence + adapter sequence
                                 adapter sequence)
```

Preparation of File Sample

3)Refer the following Figure for graphical explanation:



Number of reads with adapter and without adapter for each FASTQ file

Preparation of File Sample

4)5 type of FASTQ file was created which include the following characteristics:

- File1: 100% overlap between read 1 and read 2

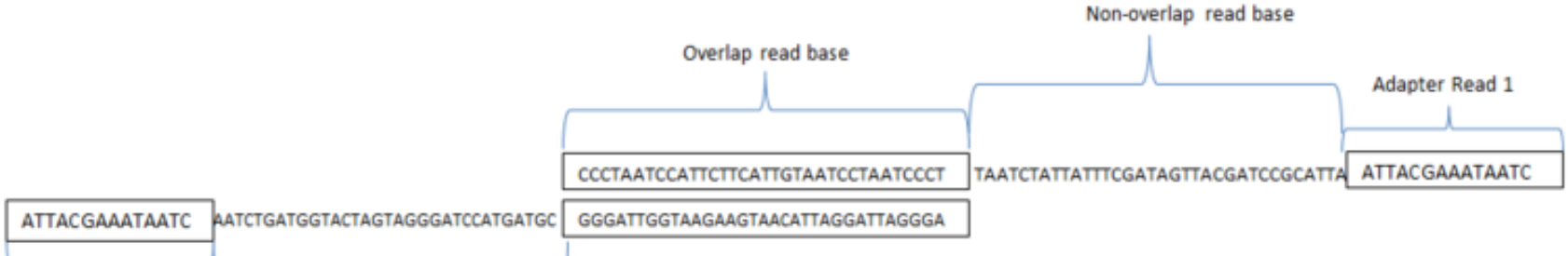
Figure 1

Preparation of File Sample

- File2: 75% overlap between read 1 and read 2

Figure 2

- File3: 50% overlap between read 1 and read 2



Preparation of File Sample

- File4: 25% overlap between read 1 and read 2

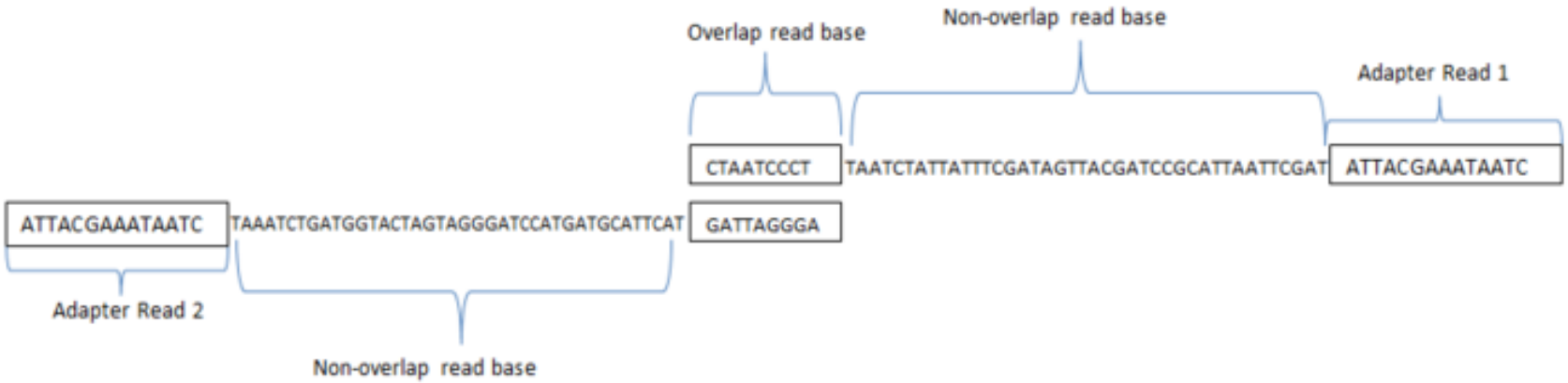


Figure 4

- File5: 0% overlap between read 1 and read 2

Output's accuracy Result

percentage of overlap sequence (%)	number of adapter detected (/20)
100	20
75	0
50	0
25	15
0	0

Code Optimization

Original code vs Edited code

original code

edited code

```
const int  s[ 4 ][ 4 ] = { { a, b, b, b },     /* Subs
                           { b, a, b, b },
                           { b, b, a, b },
                           { b, b, b, a } } ;

int  L1 = seq_1.length();
int  L2 = seq_2.length();

for( i = 1; i <= L2; i++ )
{
        for( j = 1; j <= L1; j++ )
        {
                nuc = seq_1[ j-1 ] ;

                switch( nuc )
                {
                        case 'A':  x = 0 ;  break ;
                        case 'C':  x = 1 ;  break ;
                        case 'G':  x = 2 ;  break ;
                        case 'T':  x = 3 ;
                }

                nuc = seq_2[ i-1 ] ;

                switch( nuc )
                {
                        case 'A':  y = 0 ;  break ;
                        case 'C':  y = 1 ;  break ;
                        case 'G':  y = 2 ;  break ;
                        case 'T':  y = 3 ;
                }
```

Original code vs Edited code

original code

edited code

F[i+1][j+1] = max(fU, fD, fL, ptr) ;

```
int  NW::max( int f1, int f2, int f3, char & ptr )
{
        int  max = 0 ;

        if( f1 >= f2 && f1 >= f3 )
        {
                max = f1 ;
                ptr = '|' ;
        }
        else if( f2 > f3 )
        {
                max = f2 ;
                ptr = '\\' ;
        }
        else
        {
                max = f3 ;
```

Original code vs Edited code

original code

edited code

```
while (getline (myfile,line) && getline (myfile2,line2))
{
/*
   ab.checkNucleotide(line, seq_1);
   ab.checkNucleotide(line2, seq_2);
*/
   if(line[0]=='A'||line[0]=='C'||line[0]=='G'||line[0]=='T'||line[0]=='N')
   {
        onlynuc = false;
        for(int a = 0; a < line.length(); a++)
        {
              if(line[a]=='A'||line[a]=='C'||line[a]=='G'||line[a]=='T'||line[a]=='N')
                    onlynuc = true;
              else
              {
                    onlynuc = false;
                    break;
              }
        }
        if(onlynuc == true)
        {
              seq_1 = line;
        }
   }
```

```
while (getline (myfile,seq_1) && getline (myfile2,seq_2))
{
        if(count==dnaline){
                myfile>>seq_1;
```

Result of Code Optimization

1.This table show the time taken for the programme to complete the process for each improvement's code:

Description	Change Do
	original coc
Finding Match/Mismatch score	Use if else
Finding maximum value between scoring matrix (fU/fD/fL)	use built-in

Result of Code Optimization

Description	Change Done
Filter read sequence from FASTQ file	Use count line method rather tha

Graph code change vs time

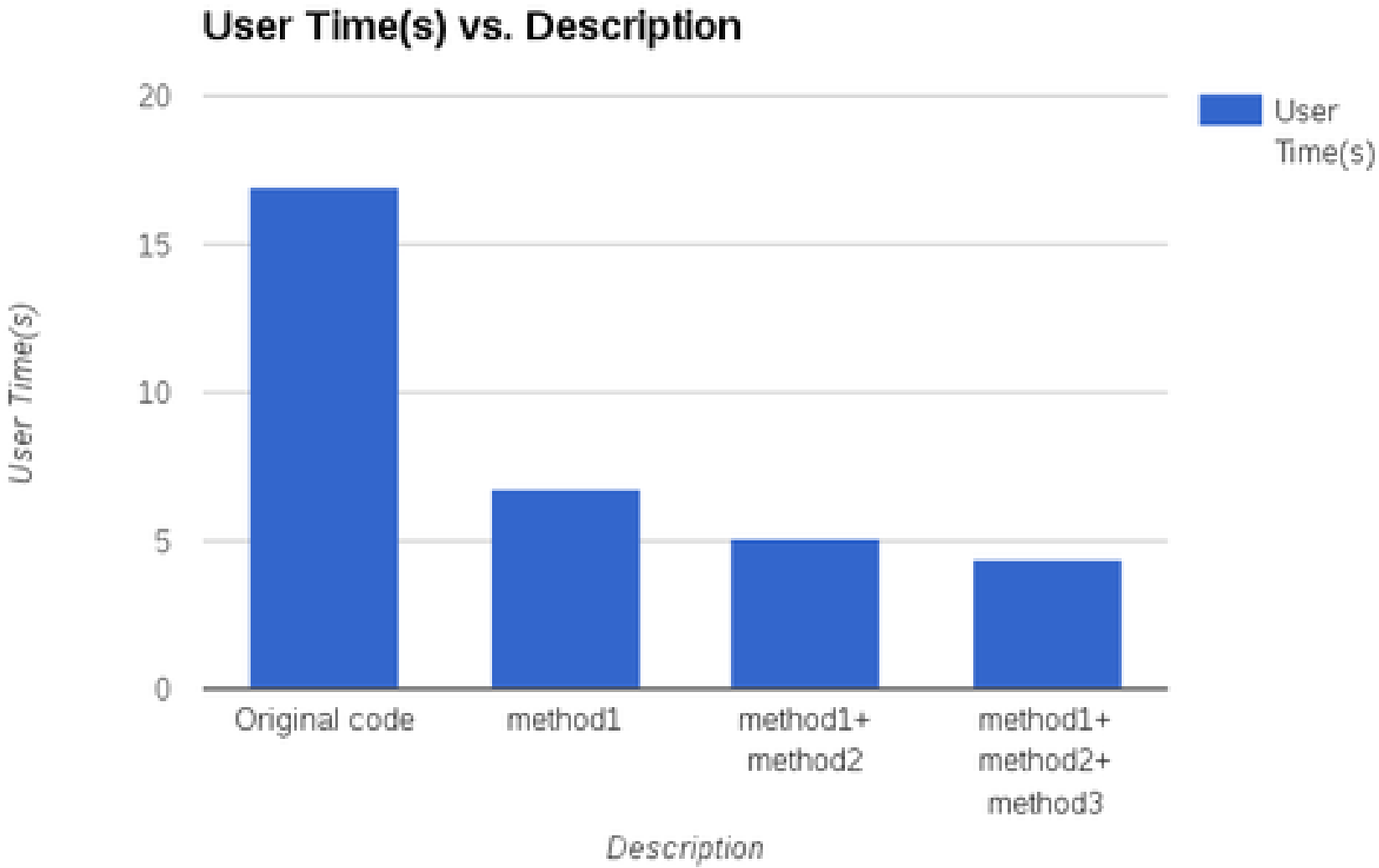


Figure 6