Julie Garcia
April 17, 2018

Lab 10

Advanced Genomics and Genetics Analyses

In this lab, we will be working with a set of unpaired test sequence reads that are 35 bp in length. There are a total of 142,858 reads in this file, which is quite small for a large-scale sequencing project. We will be comparing the assembly performance between a k-mer extension approach like VCAKE and an overlap graph-based method like Edena, and a de Bruijin graph-based method like Velvet. All 3 of these programs are very straightforward to run. The documentation for them is explicit, and if you are still uncertain of how to run the program, you can always type the program name and the arguments will be displayed to the screen.

1.) Get the file test_reads.fa from the course website.

2.) For the following 3 programs, run the arguments listed below:
- Edena with overlap size of 20 in the overlap mode and 22 in the assembly mode with depth=18X
- Velvet with a hash length of 21
- VCAKE with minimum overlap allowed = 22 (-n) and read size=35.

```
edena –r ~/Datasets/test_reads.fa –M 20
edena –e out.ovl –m 22 –d 18

./velveth velvetout 21 ~/Datasets/test_reads.fa
./velvetg velvetout

perl vcake –k 35 –n 22 –b ~/Datasets/test_reads.fa –f
~/Datasets/test_reads.fa –s contigs.fa
```

3.) How many contigs were assembled when using Edena? How many total bases were assembled into the contigs? What is the N50 value for this assembly?

```
371 contigs were assembled using Edena. 21.45 Kbp total bases were
assembled. The N50 was 57 bp.
```

4.) How many contigs were assembled using Velvet? How many total bases were assembled into contigs? What is the N50 value for this assembly?

```
16 contigs were aseembled. Total bases were 100080. N50 was 24184.
```

5.) How many contigs were assembled using VCAKE? What is the largest contig size? What is the N50? What does the distribution of contig sizes look like for the contigs assembled?

VCAKE assembled 25107 contigs, the largest being size 99995, with N50 of 35. I calculated N50 using the perl script below.

```
## perl
#calculate N50 and N90 value using a fasta file of contigs as
input
#/usr/bin/perl -w
use strict;
my ($len,$total)=(0,0);
my @x;
while(<>){
      if(/^[\>\@]/){
            if($len>0){
                  $total+=$len;
                  push @x,$len;
            }
            $len=0;
      }
      else{
            s/\s//g;
            $len+=length($_);
      }
}
if ($len>0){
      $total+=$len;
      push @x,$len;
}
@x=sort{$b<=>$a} @x;
my ($count,$half)=(0,0);
for (my $j=0;$j<@x;$j++){
      $count+=$x[$j];
      if (($count>=$total/2)&&($half==0)){
            print "N50: $x[$j]\n";
            $half=$x[$j]
      }elsif ($count>=$total*0.9){
            print "N90: $x[$j]\n";
            exit;
      }
}
```

6.) Based on this test case, which software tool is likely the least efficient, based on your best assessment? Give reasoning why.

Of the three tools, VCAKE ran the slowest (about an hour) and produced the most contigs, Velvet ran quickly and created 16 contigs, while Edena ran very quickly and produced 371 contigs. VCAKE also did not compute the N50, and an additional, albeit fast, script was needed to run. So, I would say VCAKE is the least efficient of the three, based on this test alone.