T.C.

MARMARA UNIVERSITY

FACULTY of ENGINEERING

COMPUTER ENGINEERING DEPARTMENT

CSE4074 Computer Networks

Project#1

Büşra YAŞAR-150114063

Emine Feyza Memiş-150114077

Hale ŞAHİN-150116841

INTRODUCTION

We worked on a particular region in a genome, so we only took some part of the genome as input. This input defined as one line of text in a file where the text contained only A, T, G or C. In project document, the total number of characters defined as 500 characters/bases and we allowed the total number of characters more than 500.

Our objective is to find all possible k-mers appearing at least x times. The k value will be at most 9 and x will be at least 2.

After we found all possible k-mers, searched for the reverse complement of each k-mer and when we found any, gave it as output.

ADDITIONAL INTRODUCTION

Last two days, an input file that have 540 characters has sent to us. After that, we searched some additional reverse complement of each k-mer appearing at least x times.

GENOME AND GENOMICS

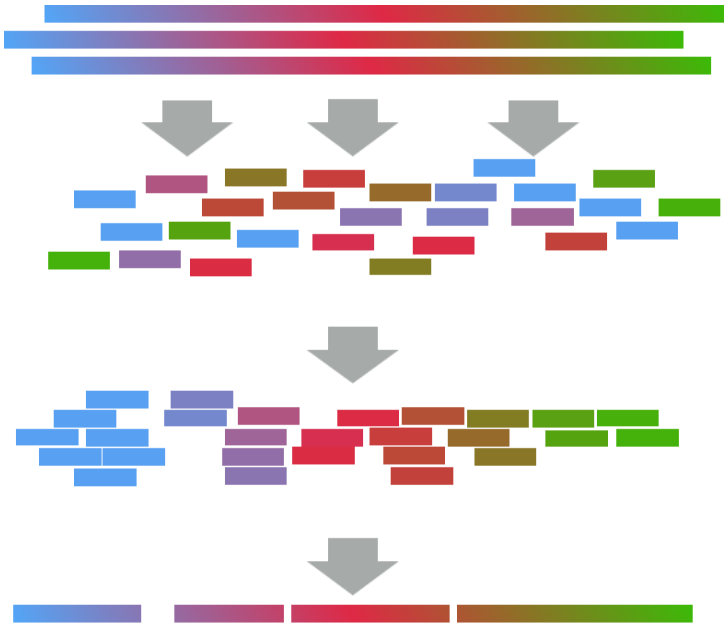
A genome is an organism's complete set of DNAs, including all of its genes. Each genome contains all the information needed to build and maintain that organism. In humans, a copy of the entire genome—more than 3 billion DNA base pairs—is contained in all cells that have a nucleus.

(Figure-1: Genome)

Genomics is the study of whole genomes of organisms and incorporates elements from genetics. Genomics uses a combination of recombinant DNA, DNA sequencing methods, and bioinformatics to sequence, assemble, and analyses the structure and function of genomes.

GENOME ASSEMBLY

The **genome assembly** is simply the **genome sequence** produced after chromosomes have been fragmented, those fragments have been sequenced, and the resulting sequences have been put back together.



(Figure-2: Genome Assembly)

DNA is double-stranded, and we have no way of knowing *a priori* which strand a given read derives from, meaning that we will not know whether to use a read or its reverse complement when assembling a particular strand of a genome.

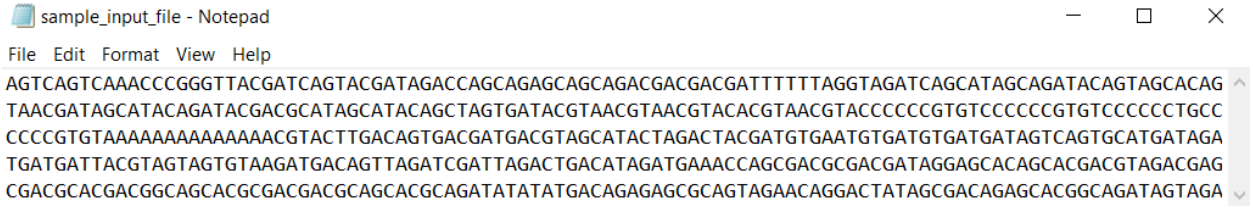
Modern sequencing machines are not perfect, and the reads that they generate often contain errors. Sequencing errors complicate genome assembly because they prevent us from identifying all overlapping reads.

Some regions of the genome may not be covered by any reads, making it impossible to reconstruct the entire genome.

Since the reads generated by modern sequencers often have the same length, we may safely assume that reads are all *k*-mers for some value of *k*.

The **reverse complement** of a DNA sequence is formed by reversing the letters, interchanging A and T and interchanging C and G. Thus, the reverse complement of ACCTGAG is CTCAGGT.

CODE AND OUTPUTS



(Figure-3: Sample Input File with the total number of characters)

```
import operator
import re
import os
import sys
from os import path
#print("Please enter the input file path :")
#path = input()
def reverse(s):
  str = ""
  for i in s:
    str = i + str
  return str

def readFile(file_path):
    path2 = path.relpath(file_path)
    with open(path2) as f:
        text = f.read()
        printText(text)

def printText(text):
    # Extract your code into a function and print header for current kmer
    #print("%s\n##############################" %name)
    kmers = {}

    for i in range(len(text) - k + 1):
       kmer = text[i:i+k]
       if kmer in kmers:
          kmers[kmer] += 1
       else:
          kmers[kmer] = 1

    print ('Inputs: k=%d,x=%d' % (k,x))
    print ('Outputs:')
    print ('%d-mer:\t' % (k))
    i = 0
    occuredKmers = {}
```

```
    passAmount = 0
    for kmer, count in kmers.items():
        if count >= x:
            print (kmer)
            occuredKmers[i] = kmer
            i = i + 1
            passAmount = passAmount + 1
    if passAmount == 0:
        print('There is no such %d-mer which appears more than %d times for the input DNA string.' % (k,x))


    print ('Reverse Complement: ')
    j = 0
    reverseComplement = ""
    noReverse = 0
    occurence = 0
    while j < len(occuredKmers):
        kmer = occuredKmers[j]
        reverseKmer = reverse(kmer)
        for base in reverseKmer:
            if base == 'A':
                base = 'T'
            elif base == 'T':
                base = 'A'
            elif base == 'G':
                base = 'C'
            elif base == 'C':
                base = 'G'
            reverseComplement = reverseComplement + base

        for i in range(len(text) - k + 1):
            kmer = text[i:i+k]
            if reverseComplement == kmer:
               occurence = occurence +1
        if occurence > 0:
            print (reverseComplement + ' appearing' + ' %d times' % (occurence))

            noReverse = noReverse + 1
        j = j + 1
        occurence = 0
        reverseComplement=""
    if noReverse == 0:
        print ('There is no reverse complement %d-mers in DNA string.' % (k))
if __name__ == "__main__":
    print("Please enter the k value: ")
    k =int(input())
    print("Please enter the x value . ")
    x =int(input())
    print("Please enter the input file path :")
    file_path = input()
    readFile(file_path)
```

(Figure-4: Codes of the Project)

```
Please enter the k value:
4
Please enter the x value .
3
Please enter the input file path :
C:\Users\Hale\Desktop\genommini.txt
Inputs: k=4,x=3
Outputs:
4-mer:
AATT
ATTT
Reverse Complement:
AATT appearing 3 times
AAAT appearing 2 times
```

(Figure-5: Output of k is 4 and x is 3)

```
Please enter the k value:
3
Please enter the x value .
4
Please enter the input file path :
C:\Users\Hale\Desktop\genommini.txt
Inputs: k=3,x=4
Outputs:
3-mer:
TTT
Reverse Complement:
AAA appearing 2 times
```

(Figure-6: Output of k is 3 and x is 4)

```
Please enter the k value:
9
Please enter the x value .
3
Please enter the input file path :
C:\Users\Hale\Desktop\genomics.txt
Inputs: k=9,x=3
Outputs:
9-mer:
ACGTAACGT
CGTAACGTA
CCCCCCGTG
CCCCCGTGT
AAAAAAAAA
Reverse Complement:
There is no reverse complement of these 9-mers in DNA string.
```

(Figure-7: Output of k is 9 and x is 3)

```
Please enter the k value:
7
Please enter the x value .
10
Please enter the input file path :
C:\Users\Hale\Desktop\genomics.txt

Inputs: k=7,x=10
Outputs:
7-mer:
There is no such 7-mer which appears more than 10 times for the input DNA string.
Reverse Complement:
There is no reverse complement 7-mers in DNA string.
```

(Figure-8: Output of k is 7 and x is 10)

```
Please enter the k value:
7
Please enter the x value .
3
Please enter the input file path :
C:\Users\Hale\Desktop\genomics.txt
Inputs: k=7,x=3
Outputs:
7-mer:
ACGATAG
GACGACG
GCAGATA
TAGCATA
AGCATAC
CGACGCA
ACGTAAC
CGTAACG
GTAACGT
TAACGTA
AACGTAC
CCCCCCG
CCCCCGT
CCCCGTG
CCCGTGT
AAAAAAA
CAGCACG
Reverse Complement:
There is no reverse complement 7-mers in DNA string.
```

(Figure-9: Output of k is 7 and x is 3)

```
Please enter the k value:
5
Please enter the x value .
5
Please enter the input file path :
C:\Users\Hale\Desktop\genomics.txt
Inputs: k=5,x=5
Outputs:
5-mer:
ACGAT
GATAG
CAGCA
GCAGA
GACGA
ACGAC
CGACG
GCATA
TAGCA
AGCAC
ACGTA
CCCCC
AAAAA
GATGA
GCGAC
GCACG
Reverse Complement:
TACGT appearing 2 times
TTTTT appearing 2 times
```

(Figure-10: Output of k is 5 and x is 5)

ADDITIONAL OUTPUTS

```
Please enter the k value:
7
Please enter the x value:
4
Please enter the input file path:
C:\Users\emine\Desktop\memiş_şahin_yaşar_project1\input.txt
Inputs: k=7,x=4
Outputs:
7-mer:
atgatca
tgatcaa
tgatcat
Reverse Complement:
There is no reverse complement 7-mers in DNA string.

Process finished with exit code 0
```

(Figure-11: Output of k is 7 and x is 4)

```
Please enter the k value:
8
Please enter the x value:
2
Please enter the input file path:
C:\Users\emine\Desktop\memiş_şahin_yaşar_project1\input.txt
Inputs: k=8,x=2
Outputs:
8-mer:
aatgatca
atgatcaa
aagcatga
agcatgat
gcatgatc
catgatca
tgatcaag
tgatcatg
ctcttgat
tcttgatc
cttgatca
ttgatcat
tgatcatc
gatcatcg
gctcttga
Reverse Complement:
There is no reverse complement 8-mers in DNA string.
```

(Figure-12: Output of k is 8 and x is 2)

```
Please enter the k value:
9
Please enter the x value:
5
Please enter the input file path:
C:\Users\emine\Desktop\memiş_şahin_yaşar_project1\input.txt
Inputs: k=9,x=5
Outputs:
9-mer:
There is no such 9-mer which appears more than 5 times for the input DNA string.
Reverse Complement:
There is no reverse complement 9-mers in DNA string.
```

(Figure-13: Output of k is 9 and x is 5)

```
Please enter the k value:
5
Please enter the x value:
6
Please enter the input file path:
C:\Users\emine\Desktop\memiş_şahin_yaşar_project1\input.txt
Inputs: k=5,x=6
Outputs:
5-mer:
atcaa
atgat
tgatc
gatca
tcttg
Reverse Complement:
actag appearing 1 times
gttct appearing 1 times
```

(Figure-14: Output of k is 5 and x is 6)

```
Please enter the k value:
5
Please enter the x value:
7
Please enter the input file path:
C:\Users\emine\Desktop\memiş_şahin_yaşar_project1\input.txt
Inputs: k=5,x=7
Outputs:
5-mer:
atgat
tgatc
gatca
Reverse Complement:
actag appearing 1 times
```

(Figure-15: Output of k is 5 and x is 7)

CONCLUSION

In this project, we used a txt file (sample_input_file) that have 500 total number of characters which consists of the genomic bases. Then, we looped over all possible starting positions of the k-mers in the text string. At each position, we extracted the k-mer by taking a slice of the string with the length of k. If the k-mer is not already in the dictionary, we added it with a count of 1. We incremented the count of this k-mer for the following each occurrence through the searching text. After that, we determined the k-mers with having

count greater than or equal to the x value. We add the corresponding k-mers to the occurred k-mers array. Then we reversed these occurred k-mers one by one and we find the complementary of the reversed k-mer. And if we find any of this reverse complement k-mer in the text file we write it out to prompt user.

ADDITIONAL CONCLUSION

After the input file that have 540 characters has sent to us, we did all the same processes again with especially 7-mer, 8-mer and 9-mer and we added new results to the report.

REFERENCES

http://claresloggett.github.io/python_workshops/improved_kmers.html

https://ghr.nlm.nih.gov/primer/hgp/genome

https://www.ebi.ac.uk/training/online/course/genomics-introduction-ebi-resources/what-genomics