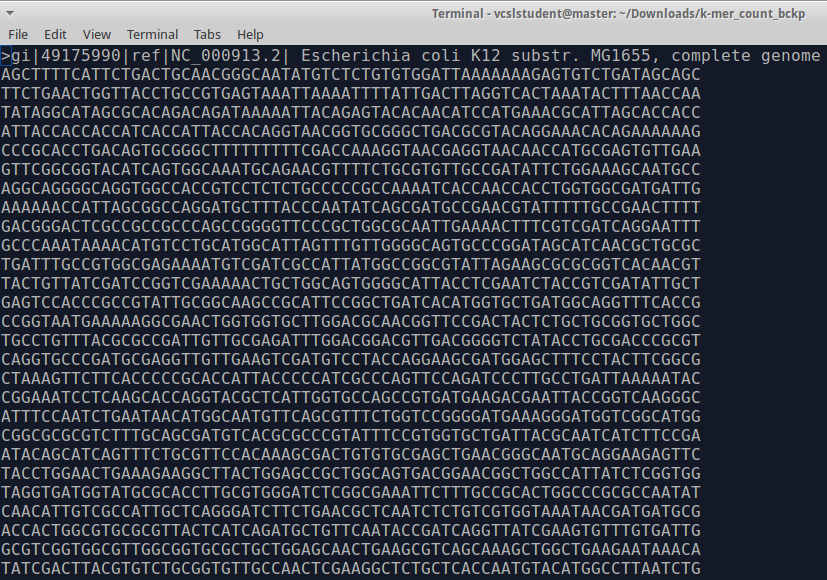
**Report**

A K-mer is a substring of length K (K > 0), and counting the occurrences of all such substrings is a central step in many analyses of DNA sequence data. Counting K-mers for a DNA sequence means finding frequencies of K-mers for the entire sequence. Counting the K-mers in a DNA sequence is a very important step in many bioinformatics applications. The following are examples of applications of K-mer counting: i) determining whether a misalignment between reads is a sequencing error or a genuine difference in sequence; ii) detecting repeated sequences, such as transposons, which are an important factor for biological role applications; iii) correcting short-read assembly errors; iv) computing metrics such as relatedness and unique enough (useful in metagenomic applications)

Although simple in principle, K-mer counting is a big data challenge, since a single DNA sample can contain several billion DNA sequences.

1. The input file given to the code is



1. The main logic used in the code to calculate the 10-mer/20-mer is as follows

*for line in sys.stdin:*

*# Ignoring the first line*

*if line[0] != '>':*

*line = line.strip()*

*# Concatenate the last line characters with this line to find 10 mers*

*line = last\_line\_chars + line*

*# iterate over each line to find 10 mers*

*for i in range(len(line)):*

*# extract 10 character string from the line*

*var\_10mer = line[i:10+i]*

*# printing only strings of length 10*

*if len(var\_10mer) == 10:*

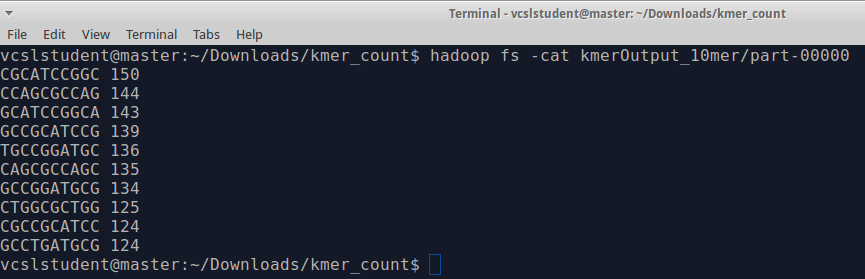
*print var\_10mer,1*

*# if the inner counter reaches length of the line - 10 then extract last 9 chars*

*if i == len(line) - 10 :*

*last\_line\_chars = var\_10mer[1:10]*

1. The 10-mer extraction output is shown as follows



And for 20-mer is as follows

