Software: K-mer counter

Genomic sequences can be viewed as continuous strings of [ATGC] that are K (K is a integer >= 1) bases long ( called K-mer, see <https://en.wikipedia.org/wiki/K-mer> for details). For example, if we would like to use 3-base long words to count the sequence ATGCATGC, we will have the following continuous 3-mers : ATG, TGC, GCA, CAT, ATG, TGC. The concept and the use of K-mer is very important in next generation sequence analysis, mostly because all sequencing reads are usually of short length (<= 150 bp). For example, the reconstruction of longer DNA sequences from short reads that we get from sequencing machine require the use of K-mer extensively, this is process is called assembly (<https://en.wikipedia.org/wiki/Sequence_assembly>). Using this technique, we can efficiently sequence any genomes and extract important information from millions of 100-150 bp long reads.

You will attempt to write your own K-mer counter using Python, which is the go-to language for prototyping programs these days in the either academia or industry. Many tech companies, including google and facebook, use python to write the majority of their programs, leaving the core parts to be written in C++ or JAVA, which is harder to write, but executes faster. For a brief read on the comparison (<http://www.differencebetween.info/difference-between-cplusplus-and-python>)

This is more of an exploratory assignment, akin to graduate level course assignment. So expect to use internet to gain a better understanding of questions asked. Since your objective is to learn, I will not worry about plagiarism.

You will use the data of two E. coli genomes, one Caulobacter genome, and one chromosome of human genome for the analysis. The human chromosome is too big to send by email, so I provide the link here (<http://www.ncbi.nlm.nih.gov/nuccore/NC_000021.9>), to download, hit “send:”, choose “File” for “choose destination”, and download in “FASTA” format. It’s about 46 million bps (so 46 MB). If you cannot download the human chromosome, then just proceed with the rest of the exercise.

These questions (comments) are here to help you understand what you are dealing with, some of them are just hints so it’s not necessary to answer them point by point. However, all programs should be implemented in a single script as different functions, this is a good habit to develop in programming.

1. Explore the .fasta files you have, what do they look like? Note the format of fasta formats, it’s very plain and simple. All annotations would go after the “>” mark, and starting from next line is the actual DNA sequence. Did you notice the differences in size? You can view them using any text editor, but keep in mind that they are relatively small in this case. Some files actually have multiple entries of “>” headed sequences, so be sure to take those into considerations when counting.
2. Implement a program/function named “ LENGTH” to count how many DNA bases there are in each file, pay attention to ignore the header “>” line. Also pay attention to the ending marks in the file by removing them.
3. Implement a program/function named “COMPLEMENT” to output the complementary strand of the DNA sequence, because DNA is duplex in nature. (COMPLEMENT.txt can be the output for example)
4. Implement a program/function to translate the DNA sequences, named “TRANSLATE”, using the standard codon table. Yes I understand that not all of the DNA in genome is coding sequence, but you can easily feed your function gene sequence to translate them.
5. Implement a program/function named “DINULC” to count the occurrences of all di-nucleotides in each of the file. For output, list the number of occurrences for each di-nucleotide and calculate their percentage. Then, use a statistical measure that you deem appropriate to compare the di-nuleotide profiles of different genomes, are they different? Are they similar?
6. Repeat the process for trinucleotide, named “TRINULC”.
7. Now, can you figure out a way to implement the K-mer counter for a user designated K? Try this on bacterial genomes as they are smaller.
8. Now you can essentially generate any K-mer profiles for each of the three genomes (+ 1 chromosome) included in this exercise. Please devise a way to compare these four chromosomes/genome and show how they are phylogenetically related.
9. What do you think the K-mer counter can be used for in bioinformatics analyses (this is an open-ended question)?