Consensus and Profile

Purpose, and Background Information

For evolutionary, it can be useful to synthesise a DNA string based on the most common nucleotide at a particular index among several homologous DNA sequences; this forms a *consensus string* which represents the DNA sequence of a likely common ancestor. An application of this methodology can be used for phylogenetic trees, and the study of evolutionary links between organisms.

A mathematical approach to visualising the theory behind this program, is with the use of a matrix. A matrix of 'x' columns, and 'y' rows; where the number of columns is equal to the length of the DNA sequence, and the number of rows is equal to the number of sequences. Such a matrix can be represented by the letter 'Axy' which returns the value at the indicated position.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | *Lengthwise DNA sequence (First loop)* | | | | |
| *Different sequences (Second loop)* | A | C | G | T | A |
| G | T | A | G | C |
| T | T | G | A | T |
| G | A | T | C | A |

*Figure 1. Table showing the arrangement of 5 nucleotide-long sequences*

A *consesus* string, as introduced before, is a string consisting of the most common nucleotide at each position. Each column within the matrix assesses the modal nucleotide, which is then added onto the consensus string.

The purpose of this program is therefore to generate a consensus string given two, or more DNA sequences of *equal* length.

Implementation

The program takes input from the user to fill a list of equal length DNA sequences.

The DNA sequences must then be analysed, and broken down into individual nucleotides at each position in the column (as explained above when discussing matrices). The idea is that the column of interest is selected first, then the sequence is selected.

To implement this, the list is first cycled via a *for* loop in the range equal to the length of a DNA sequence (here, I just chose that to be the first sequence in the list); this will select the columns in order. Once a column has been selected, and the position assigned the value *i,* we can cycle through the DNA sequences within the list (replicates the process of selecting the rows). This is achieved by extracting the elements of the DNA sequence list in a second *for* loop.

For the first pass of the first *for* loop, the value of i is zero. The value of i will be passed into the second *for* loop, and will select the first column, or the first nucleotide within a sequence (index 0). Once all the sequences have been cycled through the second *for* loop, the program returns back to the first *for* loop to move onto the subsequent column.

In order to find the modal nucleotide within a column, the index of the nucleotide stored as *i,* represents the column (as shown in figure 1, by the orange column). The nucleotides within that position are counted and stored within the variable *nuc\_dict* which is reset when moving along the columns. The most common nucleotide is found within *nuc\_dict*, and appended to the consensus string *(cat\_string*)*.* To create a matrix of nucleotides before *nuc\_dict* is erased, the counts of each nucleotide are appended to a complementary list; for example, *a\_count* for adenine. Each new appended value refers to the count for a particular column. Once the lists are printed, it creates a matrix.The loops continue until there are no more nucleotides left in the sequences.

Limitations, and Improvements

Once of the limitations of this program is that it is unable to return any more than one string. This is a problem if for a particular column, two or more nucleotides have equal counts. In such a scenario, other variations of the consensus string need to be returned. A possible solution to this problem is to set up a new class which detects when nucleotides are of equal counts. If that happens, an additional consensus string variable is initialized (such as *cat\_string\_II* as an example) to which the previous/ original one (*cat\_string*) is duplicated to up to that point. Then in the position where the equal counts occurred, the other nucleotide is appended instead. This process repeats if there are any other columns with such a scenario.