DESIGN DOCUMENTATION

This search engine helps you find similar dna sequence for a particular query in the given dataset. Here the dataset contains 1058 dna sequence, where in each line is considered as a document. Here the given query is added into dataset temporarily and will be removed after finding the similar documents. Now after adding the given query into the dataset as document, the whole data set will undergo shingling, vectorizing to form a matrix where each row is a document and each column is unique shingle. Then this matrix undergoes min-hashing to form similarity matrix. Now, we implement latent similarity hashing on this similarity matrix to give all the similar pairs of documents as tuples. Now we find the tuples containing 1 in them as our query is stored in the first line of the dataset. After displaying the results, query is removed from the dataset.

This program is divided into 5 subparts:

1. **frun.py –** Prints out the set of tuples obtained in ish.py

2. **lsh.py –** Here Gives out the list of similar sentences to query by checking the jaccard index with similarity matrix generated in minimumHashing.py part

3. **minimumHashing.py** – Creates a signature matrix of smaller size after shuffling the hashed matrix using shuffle, shuffle is used to generate a random shuffling of an array of elements 0 to size-1.

4. **shingling.py –** Creates a dictionary with key as the DNA string and the set of shingles of size as mentioned in input from the dataset linked

5. **vectorize.py –** Creates a numpy array with shingles as rows and documents as columns, the array has 0 or 1 value based on whether the shingle is present in document or not

Dataset for this program has been stored in a .txt file:

1. **dataset.txt** – Stores the inverted index that is created by inverted_index.py. Here it is stored as a dictionary where words in the vocabulary are stored as key and value of the key is a list containing all the documents in which the key has occurred.

The dataset is stored in a .txt file inside the assignment.