CEN 350 - Theory of Computation

Louis Alban Ziko

Assignment Report - Levenstein Algorithm

Checking the similarities between files

1 Abstract

This assigment involved writing a program which would find the similarities between 10 DNA files. The files were 100 characters long and were randomly generated. The program would check each of the files against each other to form a table of how similar they are to each other.

2 Introduction

Levenstein Algorithm is used for finding the difference between two strings. It finds a number which is the minimum number of changes required to change the first string into the second. The time complexity of the algorithm depends on the length of the two strings being compared. So if the lengths are m and n the time complexity is $\mathcal{O}(n * m)$ since we have to go through all the pairs of characters from the two strings.

3 Dataset

As instructed I generated 10 random DNA sequences from 'https://faculty.ucr.edu/ mmaduro/random.htm' which I then saved into 10 different files which are stored with the code under the 'files' folder.

4 Experiments and Results

4.1 Task 1

The first task involved creating the 10 files.

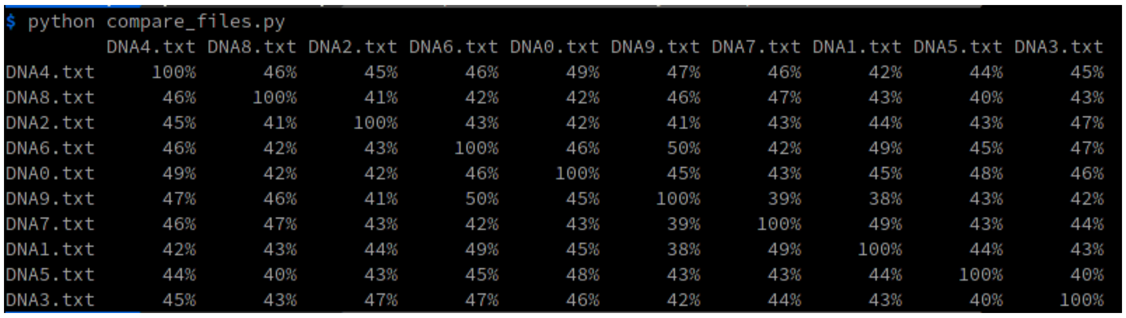
4.2 Task 2

The second task involved writing a program which utilized the Levenstein algorithm to find the similarities between the created files.

To do this I used python because it easy to use and doesn't require a compiler. Also I found an implementation of the Levenstein algorithm at the following website [1]. By using the function written I was able to pass each of the contents of the files in pairs and form a table which showed the results.

4.3 Task 3

The third task involved showing the resulting table:



```
$ python compare_files.py
        DNA4.txt DNA8.txt DNA2.txt DNA6.txt DNA0.txt DNA9.txt DNA7.txt DNA1.txt DNA5.txt DNA3.txt
DNA4.txt     100%      46%      45%      46%      49%      47%      46%      42%      44%      45%
DNA8.txt      46%     100%      41%      42%      42%      46%      47%      43%      40%      43%
DNA2.txt      45%      41%     100%      43%      42%      41%      43%      44%      43%      47%
DNA6.txt      46%      42%      43%     100%      46%      50%      42%      49%      45%      47%
DNA0.txt      49%      42%      42%      46%     100%      45%      43%      45%      48%      46%
DNA9.txt      47%      46%      41%      50%      45%     100%      39%      38%      43%      42%
DNA7.txt      46%      47%      43%      42%      43%      39%     100%      49%      43%      44%
DNA1.txt      42%      43%      44%      49%      45%      38%      49%     100%      44%      43%
DNA5.txt      44%      40%      43%      45%      48%      43%      43%      44%     100%      40%
DNA3.txt      45%      43%      47%      47%      46%      42%      44%      43%      40%     100%
```

4.4 Task 5

The fifth task involved finding the 3 most similar files to the first file. Let's take 'DNA4.txt' as the first file since it appears first in the table. Then the most similar files to it, according to the table, are:

1. DNA0.txt with 49%
2. DNA9.txt with 47%
3. DNA8.txt, DNA6.txt and DNA7.txt with 46%

References

[1] Frank Hofmann. Levenshtein distance and text similarity in python. https://stackabuse.com/levenshtein-distance-and-text-similarity-in-python/.