CS904 Assignment 1

Callum Marvell
u1606521

January 30, 2020

1 Introduction & Overview

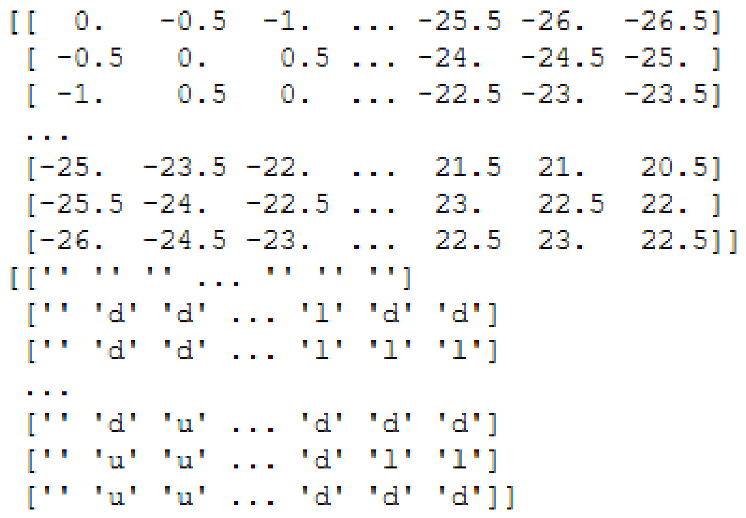
This brief report outlines the results obtained by the code written for the CS904 assignment, focused on sequence analysis.

2 Task 1

The objective of task 1 was to build an implementation of the Needleman-Wunsch Algorithm (NWA) in order to calculate the similarity score between two arbitrary RNA sequences.

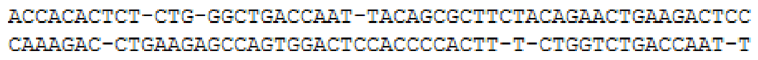
The final code implemented here produces a distance/score grid, detailing the optimal alignment score up to any given point within the two sequences, as well as a "direction" matrix, detailing the directions each cell may have been reached from when following an optimal path from the origin - thus facilitating the backtracking necessary to find an optimal alignment. This specific solution is not complete (and is not guaranteed to find all possible optimal matchings), but will always find at least 1 such matching.

For this specific task, the score grid (and accompanying direction grid) and an optimal matching found are as follows:



```
[[  0.   -0.5  -1.  ... -25.5 -26.  -26.5]
 [ -0.5   0.    0.5 ... -24.  -24.5 -25. ]
 [ -1.    0.5   0.  ... -22.5 -23.  -23.5]
 ...
 [-25.  -23.5 -22.  ...  21.5  21.   20.5]
 [-25.5 -24.  -22.5 ...  23.   22.5  22. ]
 [-26.  -24.5 -23.  ...  22.5  23.   22.5]]
[['' '' '' ... '' '' '']
 ['' 'd' 'd' ... 'l' 'd' 'd']
 ['' 'd' 'd' ... 'l' 'l' 'l']
 ...
 ['' 'd' 'u' ... 'd' 'd' 'd']
 ['' 'u' 'u' ... 'd' 'l' 'l']
 ['' 'u' 'u' ... 'd' 'd' 'd']]
```

Figure 1: Score & Direction Matrices from NWA



```
ACCACACTCT-CTG-GGCTGACCAAT-TACAGCGCTTCTACAGAACTGAAGACTCC
CAAAGAC-CTGAAGAGCCAGTGGACTCCACCCCACTT-T-CTGGTCTGACCAAT-T
```

Figure 2: An Optimal Matching from NWA

Alongside this, to aid with interpretation of the optimal full matching found (encompassing the entirety of both sequences), an image is produced highlighting the path taken when matching (cells used are yellow, others are navy).

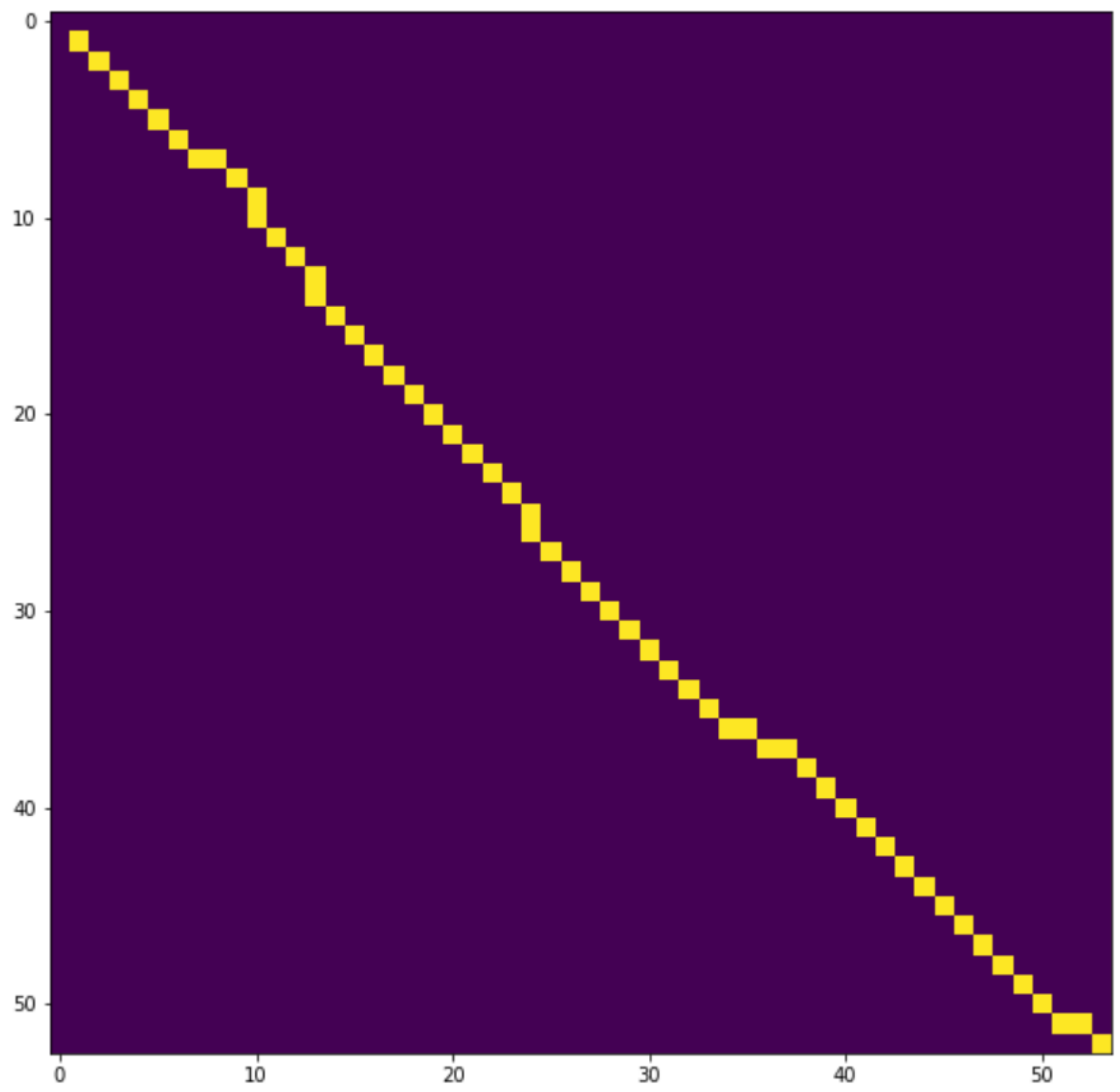
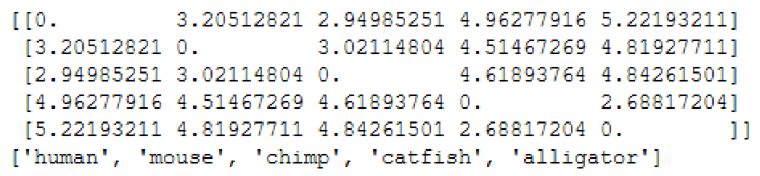


Figure 3: Optimal Path in NWA

3 Task 2 & 3

This task focused on building a phylogenetic tree using the previously developed NWA in conjunction with a neighbour joining method (NJM) implementation.

Each of the protein sequences specified was copied in FASTA format from the NCBI Entrez database. Following that, a distance matrix was formed by pairwise calculations of NWA scores and was subsequently used in the NJM algorithm as defined in lectures and online.



```
[[0.         3.20512821 2.94985251 4.96277916 5.22193211]
 [3.20512821 0.         3.02114804 4.51467269 4.81927711]
 [2.94985251 3.02114804 0.         4.61893764 4.84261501]
 [4.96277916 4.51467269 4.61893764 0.         2.68817204]
 [5.22193211 4.81927711 4.84261501 2.68817204 0.        ]]
['human', 'mouse', 'chimp', 'catfish', 'alligator']
```

Figure 4: Distance Matrix produced by NWA application

NetworkX was used to build the phylogenetic tree iteratively, and ultimately to display the final phylogenetic tree through matplotlib. Weights shown on edges correspond to the distance between nodes, where leaves are the original datapoints being compared and central nodes all have degree 3 or less. Central nodes are named as a combination of the two nodes joined by them (thus giving an idea of exactly how the phylogenetic tree was formed). Edge weights have been rounded to 2dp to aid readability, but full values are used in all calculations to prevent innaccuracy in results.

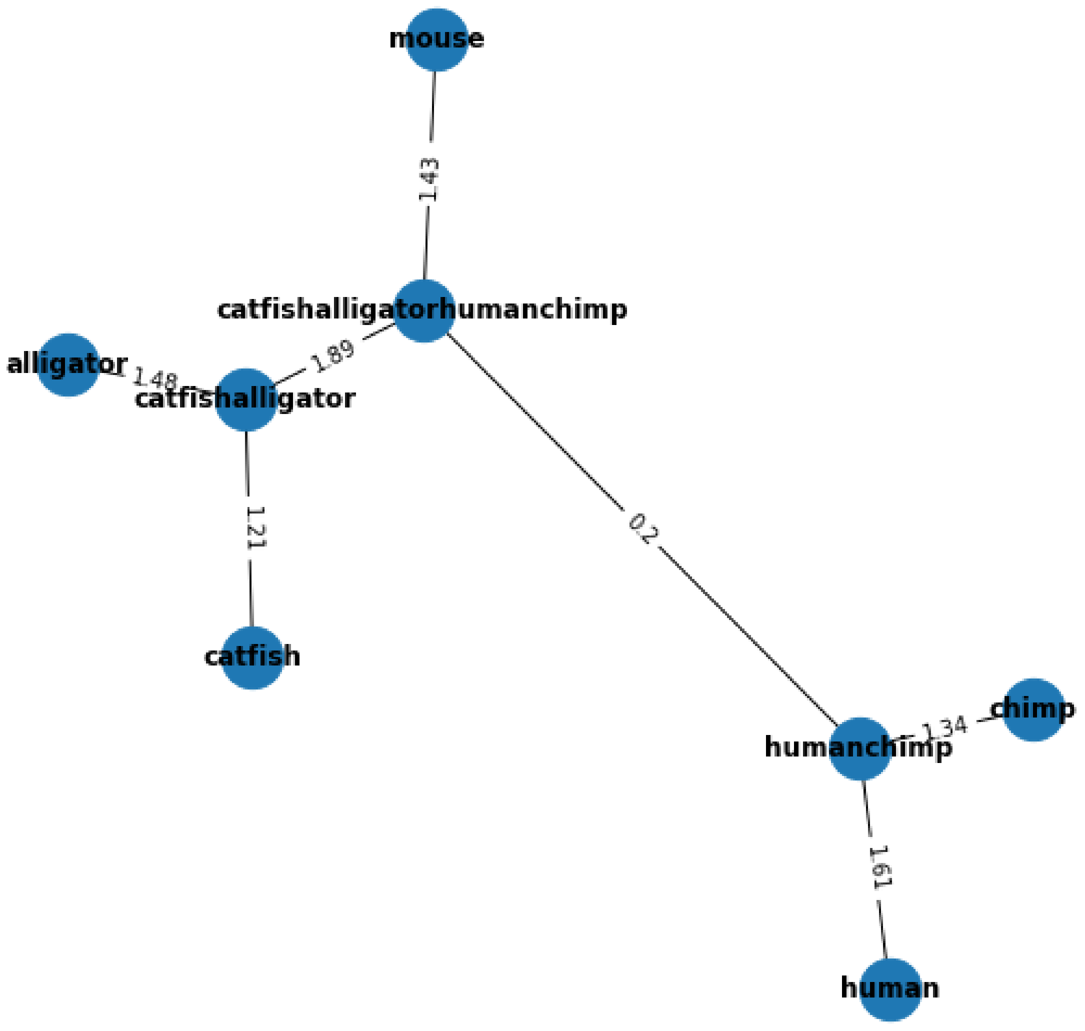


Figure 5: Example Phylogenetic Tree generated from specified taxa

Additivity of a phylogenetic tree can be analysed through brute force checking for every possible permutation of every possible subsets of leaf nodes, whether or not they fulfil the additivity requirements. If, for every subset of 4 nodes from the original nodelist (so 5C4 possibilities) has at least one valid permutation, then the phylogenetic tree can be considered additive. In this case, that is not the case and the tree is not additive.

4 Task 4

In task 4 the aim was to build a simulated six-gene toggle switch network, and appropriately analyse and model its behaviour.

To this end, 400 runs were completed over a random number of timesteps t, between 0 and 100. The data produced was dimensionality reduced from 6-dimensions to a more manageable 3 using a diffusion map approach (built ground-up from first principles, using only basic package functionality for k-means clustering from scikit-learn and various matrix mathematics from numpy). The resulting data was plotted in 3d space to illustrate the way in which the toggle-switch operates and the gene differentiates to one of the four final outcomes.

Example expected behaviour for the toggle-switch network over a single run is included below (broadly showing collapse to either one of A or B, and subsequently one of C/D or E/F respectively). Similarly, an example output from the dimensionality reduction through diffusion maps on 400 runs of random length is also shown below.

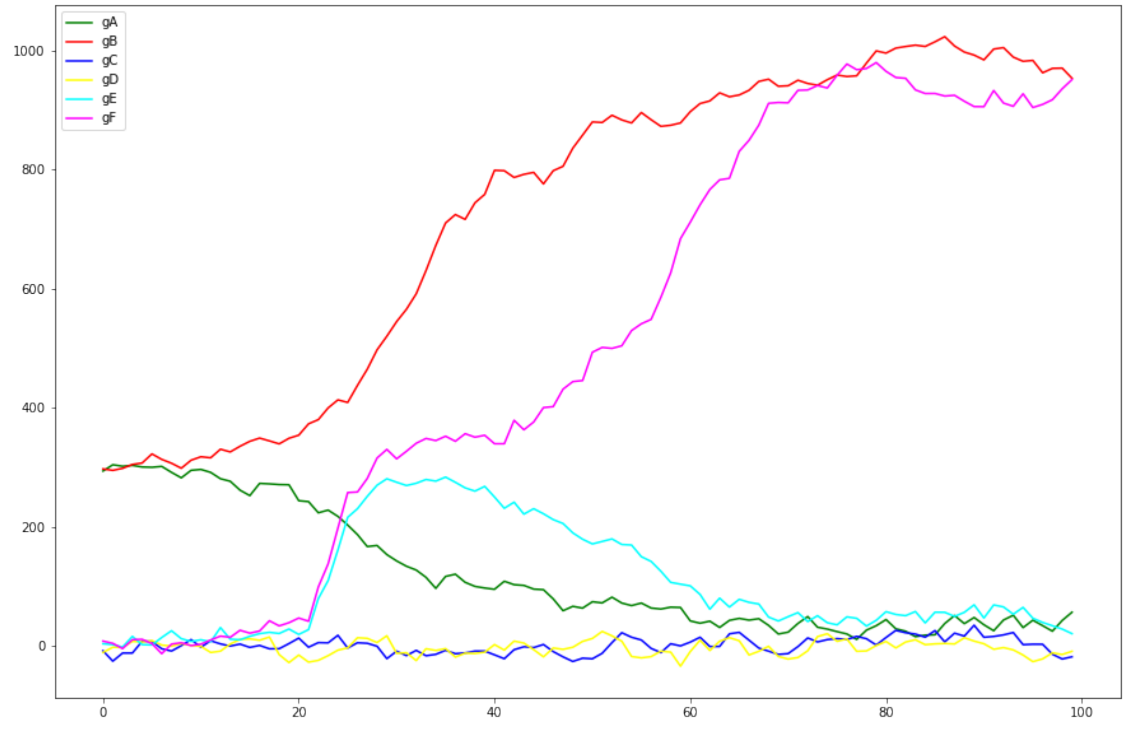


Figure 6: Single run of 100 timesteps through toggle-switch network

Similarly, the 3-dimensional data found over the 400 runs completed seems to have clustered very well into one of the possible ”collapsed states” of the original, highly unstable system. Green points are those who have not had enough timesteps to collapse to anything yet, Red points are those following collapse to gB, but not yet further differentiated to E/F, Yellow is after collapse to gF and Cyan after collapse to gE. On the other side, Blue represents those which have collapsed to gA and largely then to gC, whilst Magenta represents those which collapsed to gD. Overall, this suggests that the embedding was highly successful, allowing for most of the relevant data to be retained and communicated more easily.

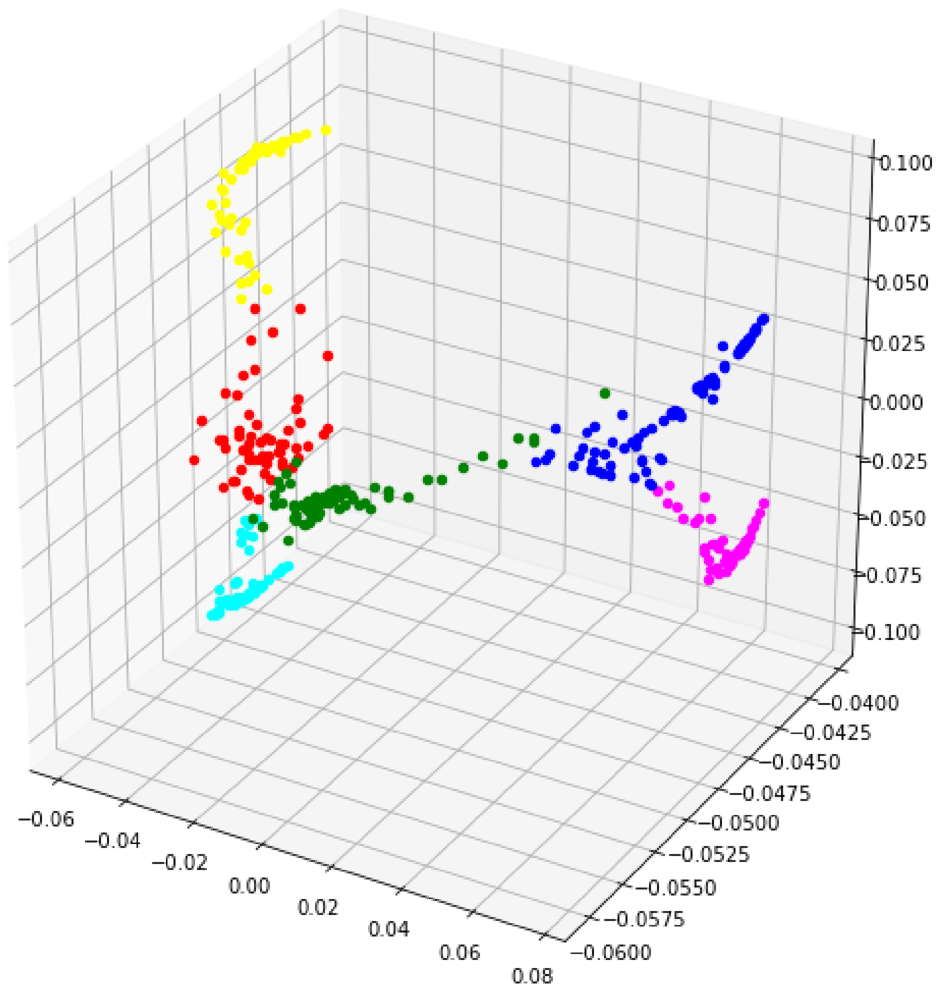


Figure 7: 3-dimensional data following Diffusion Mapping

5 Conclusion

All of the coursework tasks were completed and this report outlines the results obtained therein. Full codebases for the problems tacked in task1 can be found in Lab1_NWA.ipynb, tasks 2 & 3 can be found in Lab2_NJM.ipynb and finally task 4 can be found in Lab3_SGTog.ipynb. Checkpoints have been included for these notebooks that should still have the correct images, values, etc. loaded, but failing that execution of all cells in order should lead to comparable results.