IY3840, Coursework, 65695670

1) From the similarity graph produced by using the import address table method we see a more clustered graph produced when compared to the strings one as it has more connections. This is because with the import address table metrics we compare the DLL imports made by malware binaries so even if the initialized data section of the binaries may be unclear and anti-debugger or anti-VM anti-analysis techniques may have been used, the import declarations may have been unchanged. Therefore, by the import declarations being unchanged it could point to the reason as to why we have more connections and larger clusters. Also, with strings we look for keywords and these are more easily defined by a programmer whereas with the import address table we look at imported DLLs and function calls and so more things are looked at. The imported DLLs and function calls usually remain the same so more similarities arise.

2) From the similarity graphs produced using the instruction sequence method we see a less clustered graph compared to the strings one as the instruction sequence graphs seem to have less connections compared to the strings graph. This may be down to the fact that the instruction sequence method is known to miss many code sharing relationships between samples as the malware samples may be packed such that most of the instructions only become visible once we actually execute the malware samples and let them unpack themselves. Furthermore we are measuring similarity for the instructions method using NGrams and so we are comparing instructions by splitting them using the choice of N, this can also lead to less connections as the higher N is the less similarities are drawn as it is harder for malware samples to match. For example, in our graphs of N=3,N=7,N=11 we can see that the graphs are fairly similar due to the N value not differing by much, however N = 11 has 8 3-node connections whereas N = 7 and N = 3 both have 9. If we were to try a very large N e.g. 1500 we would have a much less connected and clustered graph than a lower number N graph, this backs up the point that the larger N is the less similarities you get.

3) From the similarity graph produced when using Dice's coefficient instead of Jaccard's index (both with threshold = 0.8) we find that the graph produced is quite clustered and has many more connections than the Jaccard's index one. This can be linked down to the fact both share different formulas. The denominator of the Dice formula is the combination of both sets whereas Jaccard's is the union of both sets. Furthermore, the numerator is the intersection of 2 sets whereas Jaccard uses the union which is avoided in the Dice Algorithm and so the Dice formula will overestimate the similarity. In terms of threshold, I went for a relatively high one (0.95). This is because this produces a similarity graph which is more accurate as the similarities between the malware are more likely to be similar. A higher threshold also leads to less false positives. This can save time in analysing malwares as less connections and clusters are present.

4) From the similarity matrix generated we can see that there is some space that is wasted as there are more plots plotted than needed. There is twice the number of plots because it is a matrix, for example malware 1 and malware 2 could be a plot on the heatmap but so would malware 2 and malware 1. Furthermore, there are plots on the map due to there being the same malware families on both the y and x axis, so the comparison between both will lead to a Jaccard index of 1. Therefore, the shape of the matrix is mostly in a diagonal going from the top left to the bottom right. The lighter colours on the matrix demonstrates a larger Jaccard index and so a stronger similarity relationship while the darker colours show a weaker similarity relationship between the malwares. Much of the matrix we see a black colour around due to the fact we are only adding Jaccard indexes over the threshold of 0.8, and so any other Jaccard index below that is added as 0. The matrix also has dark spaces in between the ones plotted on the heatmap, this could indicate false negatives as these could be missed similarity relationships.

5) Assuming I have the source code and can recompile it to generate the sample which can show me where the similarity links are, I would recommend that the printable strings all be changed to different names to each other, making it harder for them to match. Furthermore I would change all the DLL import names as well.