CS732 : Data Visualization
Technical Report for Datathon-3

Neetha Reddy
International Institute of Information Technology Bangalore
IMT2018050
Neetha.Reddy@iiitb.ac.in

***Abstract*—This technical report consists of a brief overview of methodologies employed in the graph-based visualisation of two disease-related datasets namely – Bio-diseasome and AH Sickle Cell Disease Provisional Death Counts 2019-2021. Further, this report attempts to draw inferences from the results of the above mentioned visualizations.**

I. Introduction

The Bio-diseasome dataset consists of single variate data with 512 nodes and 1188 edges of a directed graph. The Sickle cell disease dataset gives us the provisional death counts of sickle cell disease and coronavirus disease 2019 (COVID-19), by quarter, age, and race or Hispanic origin from 2019 through Quarter 1, 2021 (February 28, 2021). It contains 129 rows and 8 columns namely:

1) Data as of
2) Date of Death Year
3) Quarter
4) Race or Hispanic Origin
5) Age Group
6) SCD_Underlying
7) SCD_Multi
8) SCD and COVID-19

II. Methods and Visualizations

We have visualized all the data using *numpy*, *matplotlib*, *plotly*, *networkx*, *scipy*, *pandas*, *sklearn* and *seriate*.

A. *Single-Variate Data (Bio-diseasome)*

I have visualised the Bio-diseasome dataset using Node-link diagrams and adjacency matrix.

*1) **Node-Link Diagrams**:* I have used the *jet* colormap for all the node-link diagrams. This was done so that we can easily identify the nodes with the highest degree; this in turn might prove very useful while identifying high risk data points and drawing further inferences regarding disease causes, precautions and prevention methods. For the scope of this paper, I have covered force-directed and circular node-link diagrams.

Force-directed graphs are used to aesthetically represent network data, making it easy for us to make inferences about the subtle linkages present in the network graphs by minimising the overlap of edges in the visualisation. Fig. 1. shows Force directed Fruchterman-Reingold representation which treats edges as springs holding nodes close, while treating nodes as repelling objects and returns the final visualisation when the positions are close to an equilibrium. Fig. 2. positions the nodes according to the Kamada-Kawai path-length cost-function. From these diagrams, we observe that the highly connected nodes are pushed to the center of the graph while those with lower connectivity are pushed to the outwards.

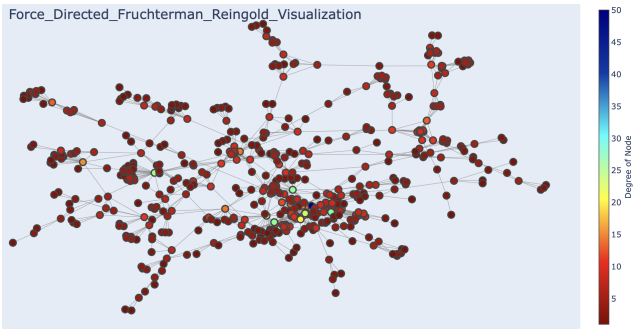

Fig. 1. Force directed Fruchterman-Reingold Node-Link diagram

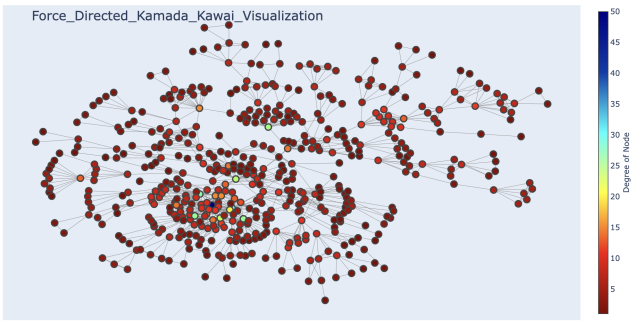

Fig. 2. Force directed Kamada Kawai Node-Link diagram

In a circular node-link diagram (Fig. 3.), the vertices of a graph are uniformly placed around the circumference of a circle. This helps us easily identify the highly connected nodes.

*2) **Matrix Visualisation**:* Matrix visualisation is a two-way, one-mode visual representation of the network, in which the nodes are ordered in rows and columns in the same way while

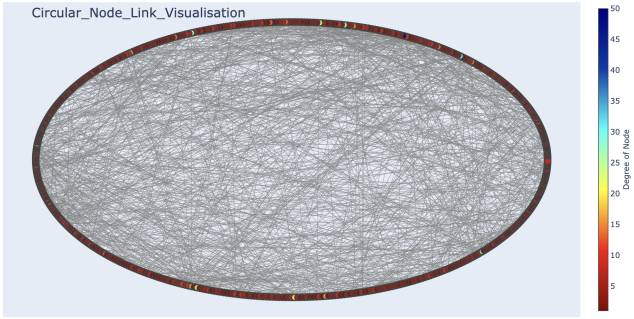

Fig. 3. Circular Node-Link diagram

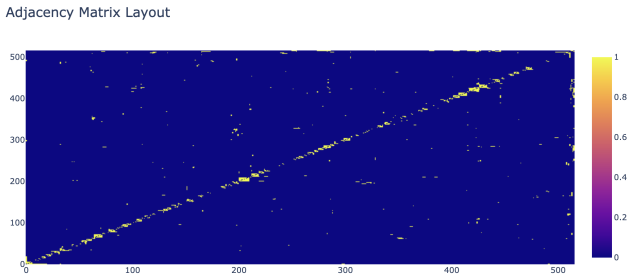

Fig. 4. Adjacency matrix

an edge is represented by colouring the block corresponding to its starting and ending nodes. This can help us detect clusters in the network graph.

B. Multivariate data (SCD death count)

I have visualised the Sickle Cell Disease(SCD) Provisional Death Counts dataset using Treemaps and Parallel coordinates plots.

*1) **Tree Maps**:* Treemaps can be used as a substitute to node-link diagrams but unlike node-link diagrams where relationship between two nodes is represented by an edge, the connections in a treemap are represented by containment. The size of the nodes are determined by the specified attributes of the node. This visualisation can be very effective when visualising data of hierarchical nature (each category having many sub-categories).

Fig. 5. Treemap of SCD_Underlying data

Fig. 6. Treemap of SCD_Multi data

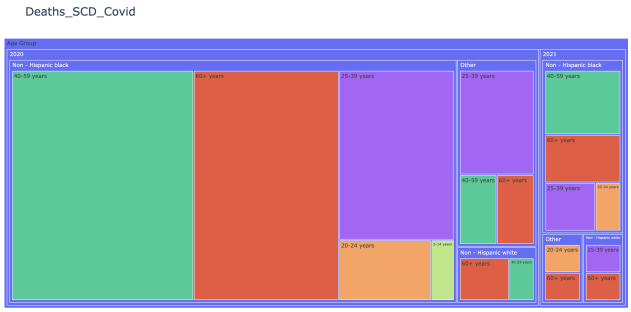

Fig. 7. Treemap of SCD and COVID-19 data

Fig. 5. represents the treemap with value SCD_Underlying and age group as the discrete color. We can see that the most affected age group is 40-59 years in all the three years followed by 25-39 years and 60+ years respectively. But 60+ years age group is more effected in 2021 in non-Hispanic black than 25-39 years age group. Further, we can observe that non-Hispanic black race is the most effected followed by Others and Non-Hispanic white race. Among Non-Hispanic whites, 60+ years age group is the most effected in both 2020 and 2019.

Trends similar to those observed in Fig. 5. are followed in Fig. 6. as well which represents the treemap with value SCD_Multi and age group as the discrete color. In Fig. 7., which represents the treemap with value SCD and COVID-19 and age group as the discrete color, we see that the year 2019 is not covered. This is because the first COVID-19 case was detected in late 2019-early 2020. Further, we can again observe that non-Hispanic black race is the most effected followed by Others and Non-Hispanic white race. In Non-Hispanic black race, 40-59 is the most effected age group followed by 60+ years age group and 25-39 years age group.

*2) **Parallel Coordinates plots**:* For these visualisations, I have label encoded the Age Group and Race or Hispanic Origin columns since they were categorical data. I have used the *sequential.Bluered* colormap.

In parallel coordinate plots, variables are plotted as axes

parallel to each other rather than the traditional orthogonal way (Cartesian coordinate system). Even the order in which the axes are placed can influence how we interpret the data. Therefore, rearranging the relative positions of the axes can help us identify patterns in multivariate data.

```
print(le1.inverse_transform([0,1,2]))
print(le2.inverse_transform([0, 1, 2, 3, 4, 5, 6]))
```

```
['Non - Hispanic black' 'Non - Hispanic white' 'Other']
['15-19 years' '20-24 years' '25-39 years' '40-59 years' '5-14 years'
 '60+ years' '<5 years']
```

Fig. 8. Label encodings

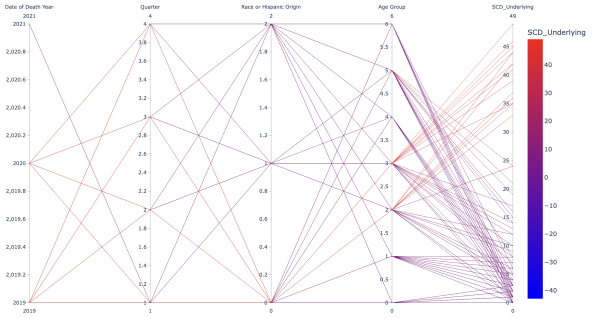

Fig. 9. Parallel Coordinate plot of SCD_Underlying data

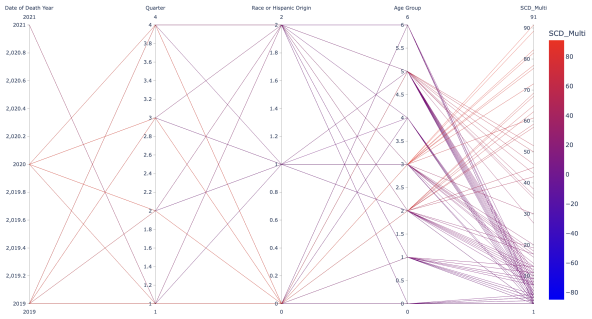

Fig. 10. Parallel Coordinate plot of SCD_Multi data

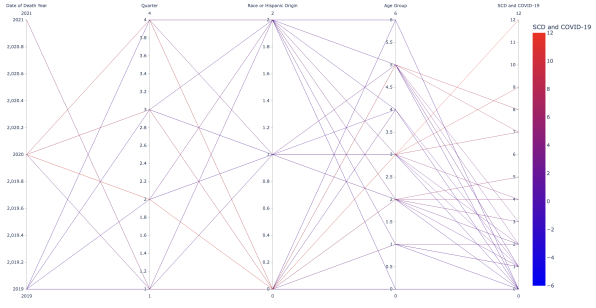

Fig. 11. Parallel Coordinate plot of SCD and COVID-19 data

III. References

1) Bio-diseasome dataset
2) AH Sickle Cell Disease Provisional Death Counts 2019-2021
3) Network Graphs in Python
4) Treemap Charts in Python
5) Parallel Coordinates Plot in Python