Bioinformatic III

Prof. Dr. Volkhard Helms
Dr. Pratiti Bhadra
Summer Semester 2021

Chair for Computational Biology Saarland University
Tutor

Exercise Sheet 4

Due: May 13, 2021 before 12:00

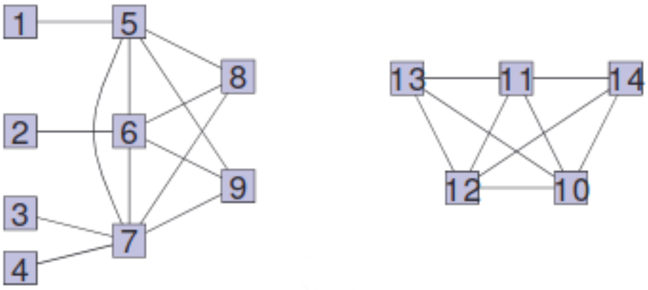
Submit your solutions to pratiti.bhadra@bioinformatik.uni-saarland.de with two attachments: (1) A ZIP file containing all your source code files, potential result files, figures and whatever else is needed to generate your solution ,(2) a PDF file containing your answers. Subject of the email should be in the following format: Assignment4-"your name".

Please feel free to contact me for any clarifications via email.

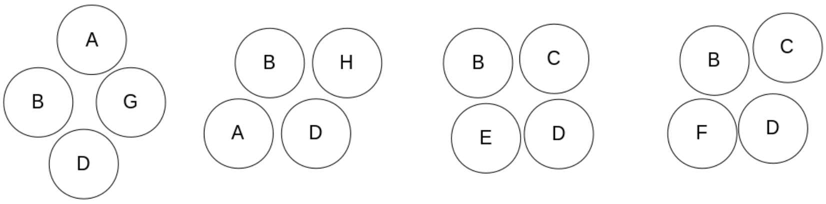
Dijkstra's algorithm, force directed layouts and modular decomposition

Q.1 Modular decomposition (10 + 15 = 25 points)

(a) Identify maximum and maximal cliques of the following disconnected graph.
A **clique** is a subset of vertices of an undirected graph G such that every two distinct vertices in the clique are adjacent; that is, its induced subgraph is complete. Cliques are one of the basic concepts of graph theory and are used in many other biological problems such as inferring protein function from protein-protein network.
A **maximum clique** is a clique of largest size. A **maximal clique** is a clique that cannot be enlarged any further.



(b) Draw the tree for the modular decomposition of this set of protein complexes and indicate the prime, parallel and serial symbols/modules.

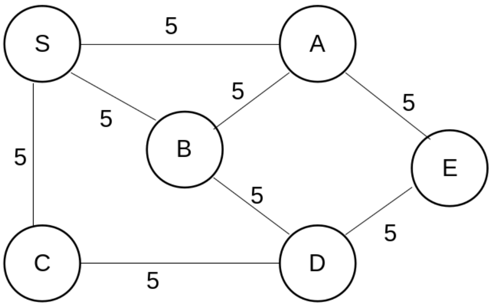


Q.2 Dijkstra's algorithm (15 + 25 + 10 = 40 points)

(a) As we know, the Dijkstra's algorithm cannot handle negative-weight edges. Is there any possibility that Dijkstra's algorithm can successfully find the shortest paths in a directed negative graph? Draw a directed graph with only one negative-weighted edge for which Dijkstra's algorithm finds the shortest path from some node s to the other nodes. What kind of change in negative weight in your graph makes Dijkstra's algorithm fail? Explain why Dijkstra's algorithm works on only the graphs with non-negative edge weights.

(b) Write python code for Dijkstra's shortest path algorithm for any graph (directed and undirected) with positive-weight edges. Document your code properly and explain the input of your code (what will be the input from the user side?).

(c) Dijkstra's algorithm is very similar to simple breadthfirst search (BFS). Which of these two algorithms will you use to find the Single Source Shortest Path from S to all nodes? Why?



Q.3 Force directed layout (15 points)

Create a layout using the "neighbour sampling" algorithm and iris.csv (given) dataset.

Install: forcelayout package in python (pip install forcelayout)

Plot the layout of iris.csv using 'forcelayout' and interpret the layout.

Neighbour sampling algorithm: A technique is presented for the layout of high dimensional data in a low dimensional space. This technique builds upon the force based methods. (Chalmers, Matthew. "A linear iteration time layout algorithm for visualising high-dimensional data." Proceedings of Seventh Annual IEEE Visualization'96. IEEE, 1996.)

Q.4 Network visualization (10 + 10 = 20 points)

Install: NetworkX (Network Analysis in Python) (pip install networkx)

(a) Plot a cycle graph of 50 nodes using the spring layout function. Represent each node using a different colour.
(hints: nx.cycle_graph, nx.spring_layout)
cycle graph: a cycle graph or circular graph is a graph that consists of a single cycle, or in other words, vertices connected in a closed chain.

(b) Plot a random graph (nx.random_geometri_graph) of 200 nodes. Color by path length from a node near the center (0.5,0.5). Identify the center.