DDA 2003/MDS 6112 Assignment 3

Start: March 6 at 9:00 AM

End: March 15 at 11:59 PM

Correspondence: Haotian Ma (haotianma@link.cuhk.edu.cn)

1 Description

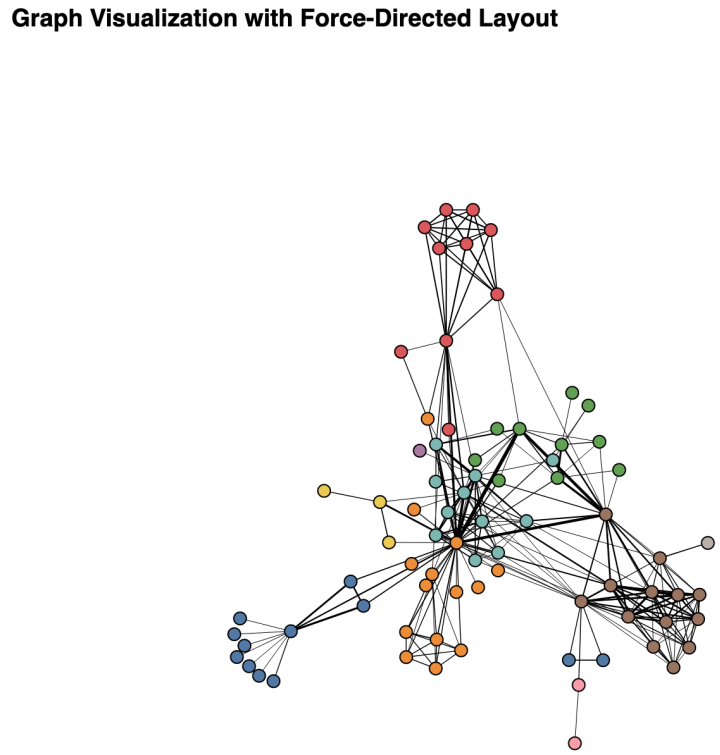
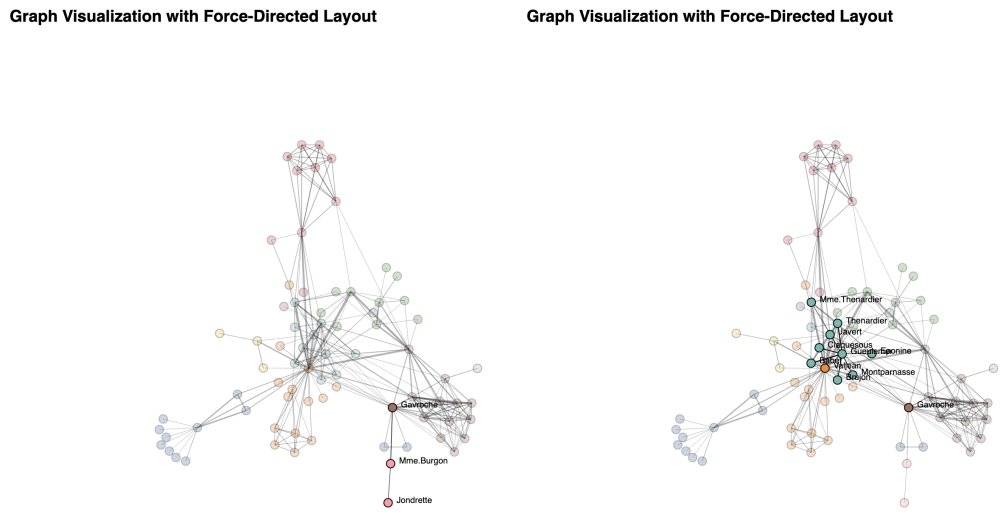


Figure 1: The visual result of assignment 3.

This assignment aims to help students get familiar with the force-directed layout in graph visualization.

In this assignment, you are asked to produce the following visualization results.



(a) (b)

Figure 2: Interaction on the graph. (a) and (b) shows that when the users hover on a node, the node and its neighborhood information will be shown. Unrelated nodes and edges will be faded.

- Given a graph stored in a .json file, visualize the graph using a force-directed layout, as shown in Figure 1.
- The color of each node is based on their corresponding groups. All nodes in one group have the same color.
- The values on edges determine the thickness of the edges. A thicker edge implies a larger value.
- When users use a mouse to hover a node, the node and its neighborhood with one order degree will be displayed, as shown in Figure 2 (a) and (b).
- Also, when a node is hovered, only the node, its neighborhood, and the corresponding edges will be shown. The rest of the nodes and edges will be faded, as shown in Figure 2 (a) and (b). "Fade" means that the opacity of the color is not 1.
- When the mouse leaves a node, all nodes and edges will be shown.

2 Requirement

- Create circles and lines for nodes and edges, respectively.
- Create a force-directed layout.
- Create a tooltip for show node information.
- Visualize graph.
- Build interaction for hovering nodes.

Online Resources The following online resources will be helpful in finishing this assignment.

- SVG in D3
- Graph visualization in D3
- Tooltips in D3
- Mouse events in D3
- Brushing and Linking in D3

Note that the students can determine the color map, the radius of nodes, and the parameters in the force-directed layout by themselves.

You are not required to use the provided template to complete this assignment. For example, you can use TypeScript as the programming language. Other programming languages, such as Python or R, are not allowed. However, you need to meet all the requirements mentioned above. In submission, ensure you offer detailed instructions so the grader knows how to produce your results successfully.

3 Instruction

3.1 Define node and link

In this part, you are asked to define node and link in a graph using D3. Node is a group element, and link is a group element in HTML. When defining the

node, you must specify its stroke, marker, and bind data. Similarly, during defining link, you need to specify stroke opacity, marker, stroke width, and bind data. Note that the stroke width of each link should be associated with the value of the link. Figure 3 shows the output of the node and link. Note that since we build interaction on the node, you need to add two mouse events, i.e., mouseover.fade and mouseout.fade, into the node variable. When these two mouse events are triggered, they will call the fade() function.

(a) Node structure (b) Link structure

Figure 3: Output of node and link.

3.2 Force-directed layout

In this part, you are asked to create a force-directed layout using d3.simulation(). You need to specify the following attributes: link, charge, and center. The output of the layout is shown in Figure 4.

Figure 4: Force-directed structure.

3.3 Set up tooltip

In this part, you are asked to define a tooltip, which will be used in building the interaction. The tooltip is also a group element in HTML. During the definition, you need to specify the text attribute.

3.4 Visualize graph

You are asked to visualize the graph using the defined force-directed layout in this part. The procedure has two steps. First, you need to add the node and link data defined in the first part to the layout. Second, set the tick attribute and provide the corresponding attributes (such as the x and y positions) to the nodes and links. Also, you need to specify the color for each node, and do not forget to add the tooltip to the force-directed layout.

3.5 Build interaction

In order to build interaction, you need to complete two functions in the assignment: fade and isConnected functions.

fade() function: The function has one parameter: opacity. The opacity value ranges from 0 to 1. When the mouse hovers, we reduce the opacity (e.g., 0.4) of all unhovered nodes. when the mouse does not hover, we set opacity as 1 for all nodes. In this function, we need to update the following attributes i.e., the opacity of the node, visibility of the text, and stroke-opacity of the link. The rule of setting up the opacity of the node and the visibility of the tooltip is based on the isConnected function. Meanwhile, the rule of setting up the stroke-opacity of the link is whether the source or the target node is in the link.

isConnected() function: This function accepts two nodes as input and judges whether these two nodes are connected. If they are connected, return true; otherwise, it returns false. Note that the two nodes passed into this function have different data types. You can print them on the console and understand how to access the desired information from these two data types. Considering the efficiency, you need to build a dictionary that stores all connected nodes. Namely, the two keys in the dictionary are two nodes. These two nodes have a link. The value in the dictionary is 1. The output of this dictionary is shown in Figure 5.

4 Evaluation

In total, there are 100 points in this assignment. A detailed evaluation is provided here.

1. Create circles and lines for nodes and links, respectively. (10 pts)

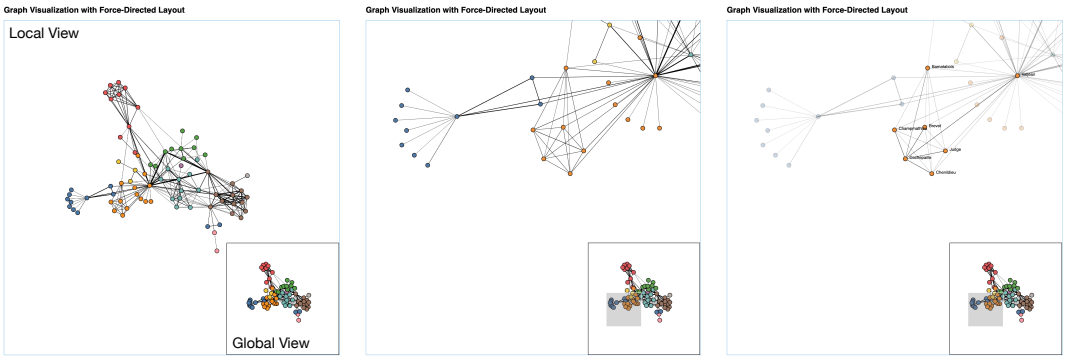
Figure 5: Dictionary structure.

2. Create a force-directed layout. (15 pts)
3. Create a tooltip for showing node information. (10 pts)
4. Visualize graph. (30 pts)
5. Complete fade function. (20 pts)
6. Complete isConnected function. (10 pts)
7. Submission (5pts). Please compress your code and a readme file (optional) into a zip file and submit the zip file to Black Board. The readme file can include descriptions that help the grader run the interface successfully.

Note that a penalty of 10 pts will be given to those students who submit the assignment one day after the deadline. A penalty of 20 pts will be given to those students who submit the assignment two days after the deadline. Submissions three days after the deadline will not be graded.

5 Bonus (50 pts)

There are an additional 50 points for the bonus. The requirements are as follows:



(a) (b) (c)

Figure 6: (a) Graph visualization with local and global views; (b) Brush the in global view; (c) hover in the local view.

- Show a global view of the graph, as shown in Figure 6 (a). (20 pts)
- User can use the mouse to brush in the global view, and the local view will update the graph based on the brushed region, sketched in Figure 6 (b). (30pts)

After brushing, users can still hover on a single node, and its neighborhood information will be displayed, as shown in Figure 6 (c).

Note that you will receive a bonus for each part only if you fully meet the requirement. Partly completing each part will receive 0 **points.**