CS732/DS732: Data Visualization -- Course Evaluation Guide

Jaya Sreevalsan Nair
International Institute of Information Technology Bangalore
October 09, 2021[1]

Grading scheme, as announced on August 3, 2021:

- 15% of the final grade for each of 4 assignments (A1, A2, A3, A4)
  - Each of the assignments has a demo for 5%, submitted as a video
- 10% for reading-writing assignment (RWA)
- 10% for mid-term
- 10% for end-term
- 10% for class attendance

Submission instructions:

- Programming assignments and RWA: 70% of the final grade
  - It is compulsory to attempt all 4 assignments and the RWA.
  - The assessment of programming assignment is based on deliverables, i.e. code+report+demo (video) for each assignment.
    - The report is where one can elaborate on the data used, hypothesis, analytical methodology, and inferences.
    - The code submissions must be source code in text format, say python code in .py text file format, and not as Jupyter notebooks.
    - The video must indicate how you have been able to demonstrate visualizations, interactions with the application, and other inferences that can be pointed out/highlighted/annotated.
  - All submissions must be done on LMS.
    - If your entire submission is larger than the permissible size for LMS submissions, upload your submission on Google Drive or Outlook OneDrive; and submit a document containing the URL to the submission.
    - It is the onus of the student to ensure the correct access permissions are provided in the repository, if the submission files reside outside of LMS, so that there is no difficulty in accessing the files for assessment. In such

[1] [An earlier versions of this document were published on August 16, September 06 -- all in 2021.]

cases, these repositories have to be accessible until the course grades are announced.

    - The scheduling of assignments provides 3 weeks to complete each of the programming assignments [A1, A2, A3], 4 weeks for A4, and 6 weeks for RWA.
        - Assignment announced by Monday midnight IST.
        - All assignment submissions are to be done by Monday midnight IST, as per schedule.
        - Assignment announcement date and submission deadline:
            - A1: Aug 16, Sep 06;
            - A2: Sep 06, Oct 04 <excluding midterm week>;
            - A3: Oct 04, Oct 25;
            - A4: Oct 25, Nov 22;
            - RWA: Oct 04 (paper selection by Oct 16), Nov 29
- 2 written exams - 20% of the final grade
    - Possibly, proctored exams during mid-term and end-term weeks.
    - Open notes.
- 1 report-writing - 10% of the final grade
    - This can be based on a research paper or a theme. It will be allocated by first-come-first-serve.
        - The topics will be published on October 04, 2021.
        - The choice of topic/paper by the student must be communicated by October 16, 2021. If not received by the deadline, a randomly picked topic/paper by the instructor by October 18, 2021.
        - The report is due on November 29, 2021.
    - If it is based on a research paper, the report must say why the method is important, and its impact on the research community (using the papers that have cited the paper, a state-of-the-art paper that explains the value of the paper), etc.
    - The technical report is to be written in the IEEE conference paper format.
    - This also includes references to papers/articles/etc. by citing them appropriately in-place in the report. There will be negative points for not doing citation references within the article and bibliography properly.

A1:

Date of announcement: 11:59 pm IST, August 16, 2021 (Monday)
Date of submission: 11:59 pm IST, September 06, 2021 (Monday)
Summary: The assignment is on 2D scalar and vector field visualization.
Dataset: IEEE Visualization 2008 Design Contest dataset
(http://sciviscontest.ieeevis.org/2008/data.html)

The dataset description is given on the dataset website. The data files are available at https://cloud.sdsc.edu/v1/AUTH_sciviscontest/2008/data_files/

The scalar field datasets are multifield.xxxx.txt.gz and vector field as velocity.xxxx.txt.gz, where xxxx refers to the timestep. There are 200 timesteps.

Tasks:

1. Choose sufficient timesteps to visualize the progression of the simulation. Use the same time-steps for both multi-field and velocity files.
    - Write in the assignment report -- how did you arrive at the timesteps? Data-driven methods are encouraged.
2. For the files chosen, identify one 2D plane you will be studying. The 2D plane can be the x-y plane at a constant z value, the y-z plane at constant x, and the z-x plane at a constant y value.
    - Write in the assignment report -- how did you arrive at which plane you are going to explore?
3. For the multi-field files, choose 3-5 scalar fields you will be studying.
    - Write in the assignment report -- what is your rationale for the selection of variables?
4. For the vector field, use the curl as the vector field to visualize, as given in the data description webpage.
5. Outputs:
    - For scalar field visualization, use color mapping and contour mapping (or contour fill) for 5 contours; for vector field visualization, use quiver/arrow plots. Address the following in your report:
        - For contours, will you use the same contour values for all time steps?
        - For color-mapping, will you use the same min-max values to generate the color palette?
    - Experiment with different types of color palettes/spectrum (sequential, diverging, qualitative) using colorbrewer/matplotlib predefined palettes.
        - Write in the report - did any color palette outperform the others? How would you rationalize the performance?

- Experiment with combining 2 visualization techniques in a single view.
    - Did such a visualization enable you to make joint inferences of different fields?
- Generate demo videos of the animation of change in scalar and vector fields over time.
- Write in your report -- do your visualizations help you infer the shadow instability as shown in the image in the contest website (and Fig.1)? What in your choice of the data for visualization allowed you to see the moving front?

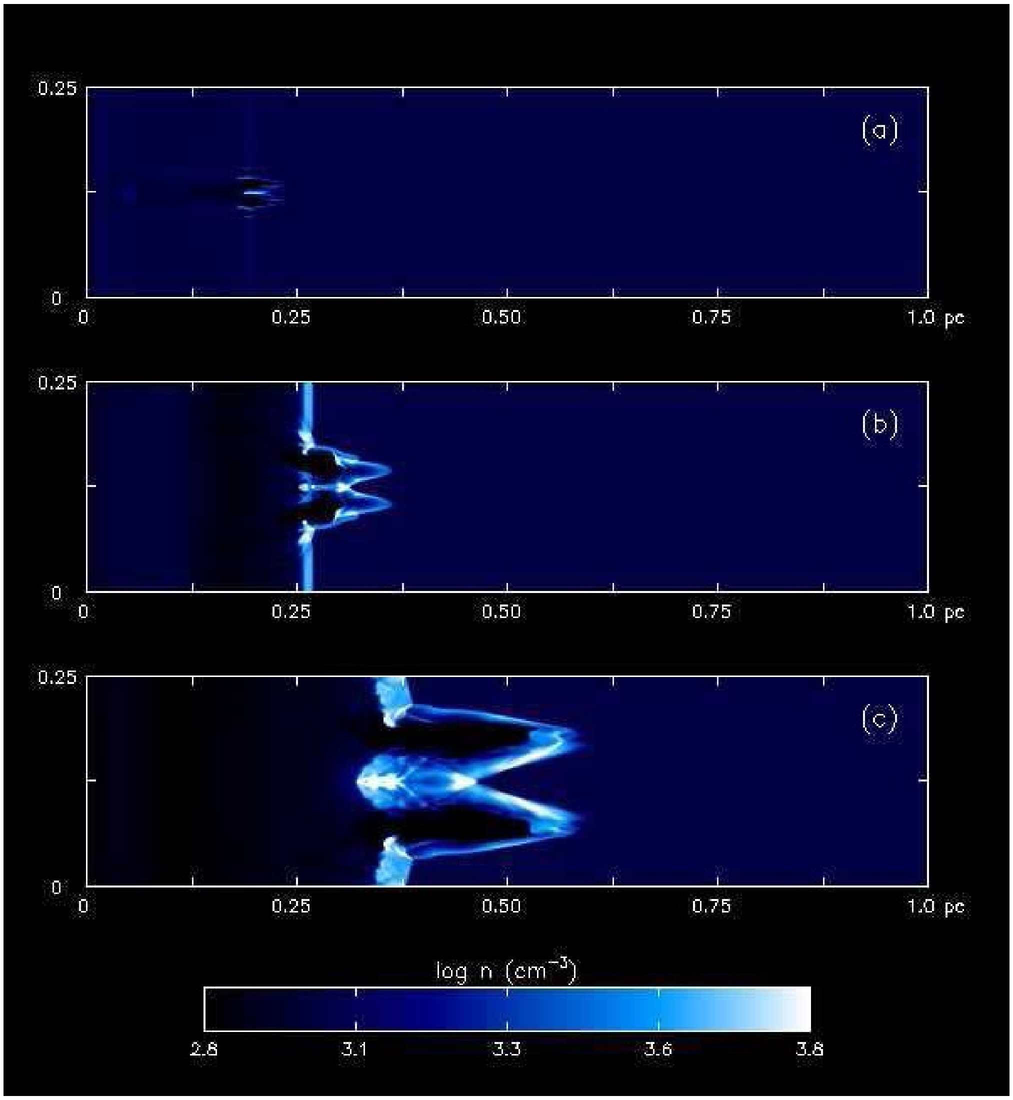


Figure 1: A shadow instability forming in one 2D slice through the data set over time (Source: http://sciviscontest.ieeevis.org/2008/)

A2:

Date of announcement: 11:59 pm IST, September 06, 2021 (Monday)
Date of submission: 11:59 pm IST, October 04, 2021 (Monday)
Summary: The assignment is on 3D scalar and 2D vector field visualization.
Dataset: IEEE Visualization 2008 Design Contest dataset
(http://sciviscontest.ieeevis.org/2008/data.html)

The dataset description is given on the dataset website. The data files are available at https://cloud.sdsc.edu/v1/AUTH_sciviscontest/2008/data_files/

The scalar field datasets are multifield.xxxx.txt.gz and vector field as velocity.xxxx.txt.gz, where xxxx refers to the timestep. There are 200 timesteps.

Tasks:

1. For the 2D vector field visualizations, generate streamline visualizations and compare the quiver plots generated in A1 with streamlines.
    - What are your inferences from the comparison of the visualizations with respect to the effectiveness of visualization?
2. From your experience in A1, identify a volumetric dataset at a specific timestep and for one of the scalar fields, select 5 isosurface values, and perform isosurface extraction.
    - Explain your choice of the volumetric dataset (time-step and the scalar field), and that of the isosurface values.
    - Experiment with transparencies for displaying the 5 isosurfaces simultaneously. What are your learnings? Did the use of transparency improve the visualization? Did the data have isosurfaces as layers that could exploit the use of transparency for improving visualization?
3. Using the volume and scalar field used in #2 in A2, perform oblique slicing. Your program should take 4 values for the plane equation as an input. The slicing plane must be able to move along its axis, i.e. its normal vector through any point on the plane. The visualization output can be presented as a 2D color-mapped image (Fig. 2, right), similar to the outputs in A1; or it could be the visualization of the slice in-situ in the 3D volume (Fig. 2, left).

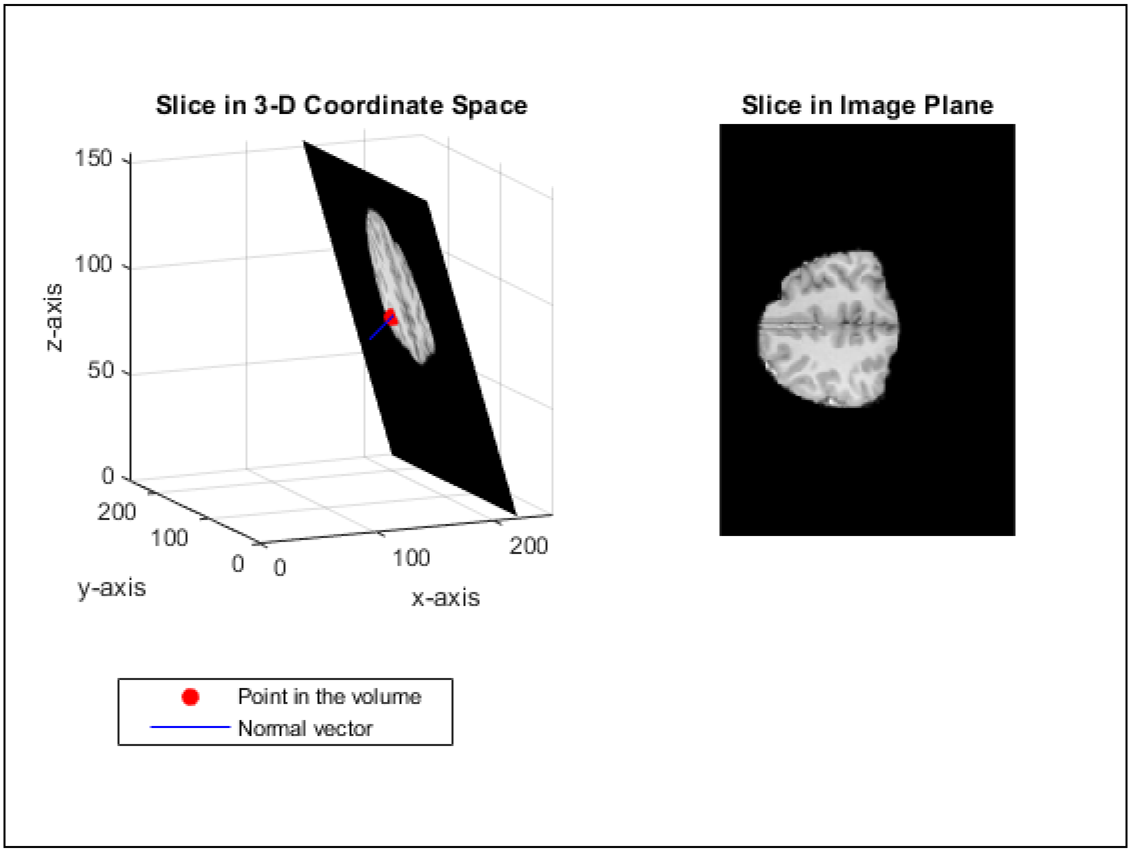


Figure 2: Outputs of oblique slicing in 3D volumes -- (left) in-situ in the 3D volume, and (right) color-mapped image of the cross-sectional view. (Source: Mathworks)

- What is the procedure you have used to make the slicing plane slide along its axis, i.e. the plane normal?
- How did you compute the intersection of the slice with the volume?
- How is the scalar value computed for the points in the volume, that are not on the grid, for generating the color of the cross-sectional surface?

A3:

Date of announcement: 5:00 pm IST, October 09, 2021 (Saturday)
Date of submission: 11:59 pm IST, October 25, 2021 (Monday)
Summary: The assignment is on browser-based visualizations of networks, trees, and multivariate data.

Libraries to be used: Plotly or D3.js
Diseases related datasets:
* Network data - diseasome: https://networkrepository.com/bio-diseasome.php
* Network data - : https://networkrepository.com/infect-dublin.php
* Multivariate data: https://catalog.data.gov/dataset/ah-sickle-cell-disease-provisional-death-counts-2019-2020-3e71c
* Multivariate data: https://catalog.data.gov/dataset/u-s-chronic-disease-indicators-cdi

Visualizations to generate:

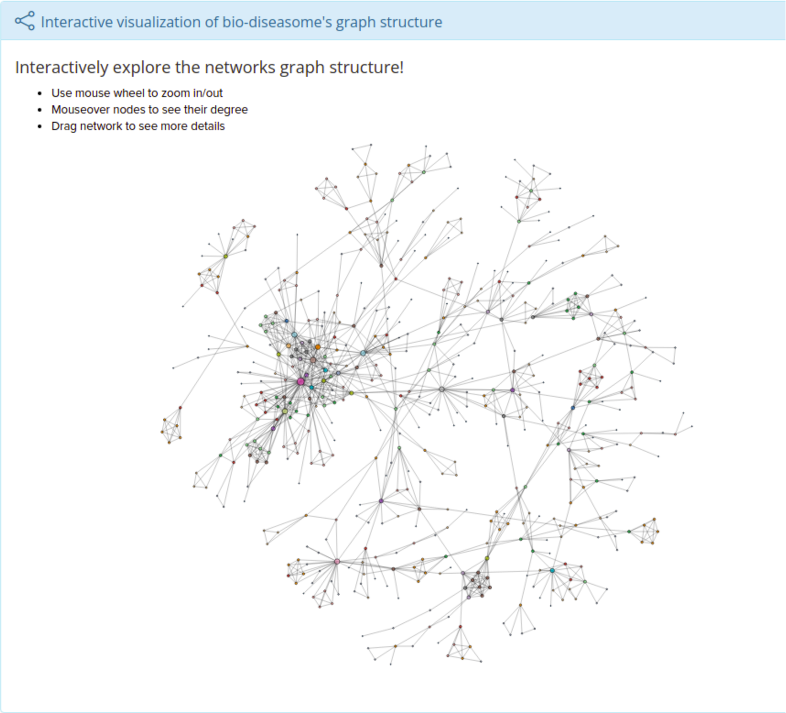
1. Node-link diagrams with two different layouts, namely, force-directed, circular, including node-labels
2. Matrix visualization of the node-link diagram generated in #1, including node-labels
3. Treemap
4. Parallel coordinates plot with axis swapping and brushing.

Tasks:

1. Pick at least one network and one multivariate dataset from the aforementioned list.
2. Decide how you would like to use the dataset for visualization -- directly or after remodeling.
3. Prepare datasets for generating visualizations in the aforementioned list, such that the prepared datasets include at least a network, a tree, and a table.
4. Implement the visualizations given in the aforementioned list.

Notes:

- The network dataset webpages in the networkrepository contain node-link diagram visualizations. These could be used as references for the visualizations you generate. Fig.3 is the reference visualization of bio-diseasome and Fig.4 is that of infect-dublin.

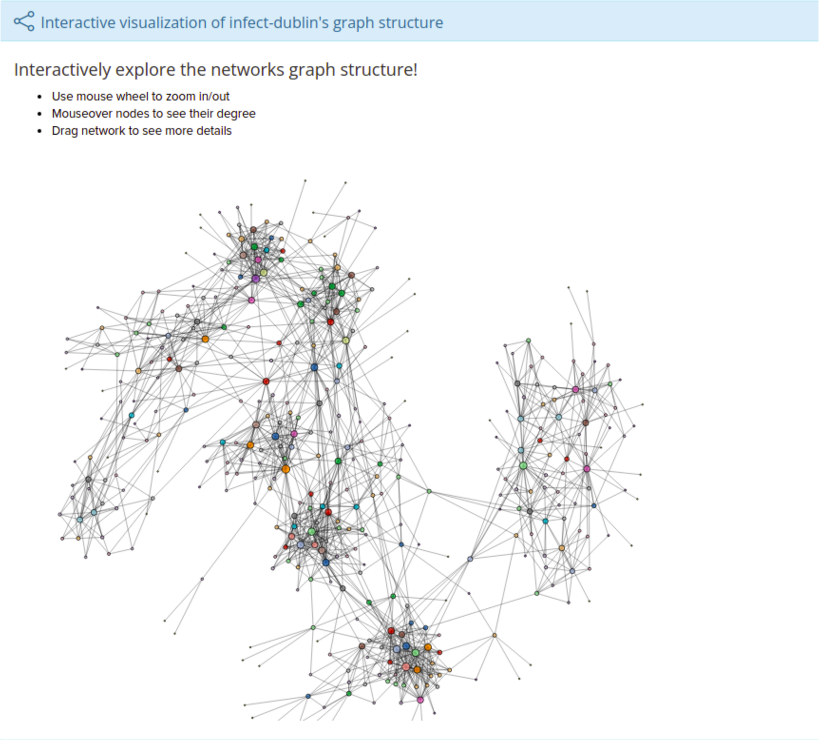


Interactive visualization of bio-diseasome's graph structure

Interactively explore the networks graph structure!

- Use mouse wheel to zoom in/out
- Mouseover nodes to see their degree
- Drag network to see more details

Figure 3: Reference node-link diagram of bio-diseasome (Source: NetworkRepository)



Interactive visualization of infect-dublin's graph structure

Interactively explore the networks graph structure!

- Use mouse wheel to zoom in/out
- Mouseover nodes to see their degree
- Drag network to see more details

Figure 4: Reference node-link diagram of infect-dublin (Source: NetworkRepository)

- The network datasets are labeled using numerical indices. Hence, these indices may be used as-is for the visualization tasks.
- The tabular data for chronic diseases contain several irrelevant columns. Pick columns based on the semantics of the variable, and known data type (categorical or numerical), as not all the variables are needed to be plotted. The same applies to the tabular data on sickle cell anemia, too.
- For remodeling, other data mining methods such as distance matrix, similarity matrix, ordering for building hierarchy, etc. may be used.
- These datasets are considered small datasets. Hence, use all rows in the multivariate datasets, and all nodes and edges in the network datasets for visualizations, unless any of this is spurious data. This implies that subsampling these datasets for visualizations will not be entertained.
- The visualizations must be rendered on a browser, hence, appropriate libraries must be used to generate HTML pages for these visualizations.