- This homework is due at 11:59pm on April 22, 2022. Please submit by email to `natalies@cs.unc.edu+comp790`.
- There are a few files provided:
  - Protein-Protein Interaction Network 1, with edges determined according to co-expression in `Coexpress_Edges.csv`
  - Protein-Protein Interaction Network 2, with edges determined according to experimental information given in `Experimental_Edges.csv`
- You are welcome to consult with other colleagues, but please write up your own independent solution.
- You are welcome to use Python, Julia, or R here.
- You are welcome to write up your assignment using the `HW2_790-166.tex` template, or write up the solutions in the method of your choice.
- This homework is worth 62 points total.
- Please submit your final writeup as a PDF. Please try not to end me pages of output from Jupyter notebooks :) I simply want to see the few lines of code you used to answer each sub-question.
- Make sure to comment and elaborate on your answers in places where you are asked to comment!

Problem 1

Understanding the Rayleigh Ritz Theorem (12 points)
Here we will empirically explore the Rayleigh Ritz Theorem which says the following.

- Let $\mathbf{A} \in \mathbb{R}^{N \times N}$ be a square symmetric matrix with eigenvalues $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ and corresponding eigenvectors $\{\mathbf{v}_1, \mathbf{v}_2, \ldots \mathbf{v}_n\}$. Defining $R_{\mathbf{A}}(\mathbf{x}) = \frac{\mathbf{x}^T\mathbf{A}\mathbf{x}}{\mathbf{x}^T\mathbf{x}}$, then the minimum value of $R_{\mathbf{A}}(\mathbf{x})$ is λ_1 and occurs when $\mathbf{x} = \mathbf{v}_1$.
- Obviously, this is a nice property to understand, as $\mathbf{x}^T\mathbf{A}\mathbf{x}$ represents the quadratic form that we are often trying to minimize.
- This property can be extended to find the matrix $\mathbf{X}$ that minimizes trace$(\mathbf{X}^T\mathbf{L}\mathbf{X})$. Specifically, the k-dimensional matrix, $\mathbf{X}$ that minimizes trace$(\mathbf{X}^T\mathbf{L}\mathbf{X})$ is the first k eigenvectors of $\mathbf{L}$ (e.g. those corresponding to the k smallest eigenvalues) and the minimum value obtained for trace$(\mathbf{X}^T\mathbf{L}\mathbf{X})$ will be $\lambda_1 + \lambda_2 + \cdots + \lambda_k$.

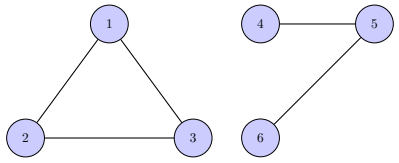

Figure 1: The graph, $\mathcal{G}$ that we already met in homework 1.

1. **Problem Setup 1** (1 point) Create the adjacency matrix, $\mathbf{A}$ for $\mathcal{G}$ and compute the graph Laplacian matrix, $\mathbf{L}$ for this graph. Note that you already did this in homework 1. Just copy it here!

2. **Problem Setup 2** (1 point) Find the eigenvalues and eigenvectors of $\mathbf{L}$. Note that you also have done this in homework 1.

3. **Eigenvalue Sorting** (2 points). We first need to order the eigenvalues of $\mathbf{L}$ from largest to smallest. That is, you need to return the indices of eigenvalues that would sort them from smallest to largest.

4. **What Won't Be the Minimum** (2 points). Find the fourth smallest eigenvalue, λ_4 and its corresponding eigenvector $\mathbf{v}_4$. Evaluate $\mathbf{v}_4\mathbf{L}\mathbf{v}_4^T$ and write down the number that you get.

5. **Compare to the Following, which will be smaller** (2 points). Find the first smallest eigenvalue, λ_1 and its corresponding eigenvector $\mathbf{v}_1$. Evaluate $\mathbf{v}_1\mathbf{L}\mathbf{v}_1^T$ and write down the number that you get. Comment on this wrt what you got using the fourth eigenvalue/eigenvector.

6. **Forming the Non-Optimal 2d Embedding** (2 points). Now we will use the eigenvectors that minimize trace($\mathbf{X}^T\mathbf{L}\mathbf{X}$). Form a matrix with two columns where the first column is the third eigenvector $\mathbf{v}_3$ and the second column is the fourth eigenvector $\mathbf{v}_4$. Define the embedding matrix, $\mathbf{E}$ as the matrix that horizontally concatenates $\mathbf{v}_3$ and $\mathbf{v}_4$ as $\mathbf{E} = [\mathbf{v}_3|\mathbf{v}_4]$.

   - Compute trace($\mathbf{E}^T\mathbf{L}\mathbf{E}$). Record what you get.
   - Compute $\lambda_3 + \lambda_4$ and comment about what you get with respect to the trace you just computed.

7. **Forming the Optimal 2 Embedding** (2 points) Now we will use the eigenvectors that minimize trace($\mathbf{X}^T\mathbf{L}\mathbf{X}$). Form a matrix with two columns where the first column is the first eigenvector $\mathbf{v}_1$ and the second column is the second eigenvector $\mathbf{v}_2$. Define the embedding matrix, $\mathbf{E}$ as the matrix that horizontally concatenates $\mathbf{v}_1$ and $\mathbf{v}_2$ as $\mathbf{E} = [\mathbf{v}_1|\mathbf{v}_2]$.

   - Compute trace($\mathbf{E}^T\mathbf{L}\mathbf{E}$). Record what you get.
   - Compute $\lambda_1 + \lambda_2$ and comment about what you get with respect to the trace you just computed.

Problem 2

(50 Points Total) **Protein-Protein Interaction (PPI) Graph Alignment**

Two protein-protein interaction networks for Humans were downloaded from the string database `https://string-db.org/` and pre-processed to produce sufficiently large but not too large subgraphs. In the first graph, edges were determined according to co-expression information `Coexpress_Edges.csv`. In the second graph, edges were determined according to validated, experimental information. Our challenge is to apply a graph alignment technique to see if the same proteins map to each other between these two graphs.
Recall REGAL alignment `https://arxiv.org/pdf/1802.06257.pdf`. The following homework sub-problems will **walk us through implementing the REGAL graph alignment approach**.

1) **Constructing Node Features (5 points):** The first part of REGAL is to create a feature vector for each node that helps to summarize something about its context. We will use a simple k-hop method to construct a feature vector for each node. Recall that for a node, i, its 'k-hop subgraph' can be obtained by considering nodes that are within k hops from i. (Hint: you may find the following useful `https://networkx.org/documentation/stable//reference/generated/networkx.generators.ego.ego_graph.html`).

We will consider k-hop networks for $k = 1, 2, 3, 4$. **Write a function, where for a particular k, you collect the set of neighboring nodes within k hops of each node and summarize the degree distribution of these collective 'k-hop neighbors' with 4 statistics : {min degree, median degree,**

mean degree, max degree}. After doing this for each value of k, you should ultimately be able to represent each node with 16 features (4 considered hops $\times$ 4 summary statistics per hop). As an example, assuming Graph 1 has N_1 nodes, define its node feature matrix, $\mathbf{X}_1 \in \mathbb{R}^{N_1 \times 16}$ matrix.

2) **Intuition Building (5 points):** Use your new function to build the described feature vectors for Supragingival Plaque Network (Network 1). Assuming this network has N_1 nodes, **project these N_1 nodes into two dimensions using your dimensionality reduction method of choice**, based on the 16 computed features ($\mathbf{X}_1 \in \mathbb{R}^{N_1 \times 16}$).

3) **Choosing Landmarks (5 points):** Recall that REGAL constructs an embedding for each node by specifying landmark nodes that have been collected across both of the graphs being aligned. Choose a set of d landmark nodes **collectively** across Graphs 1 and 2. You can play with d later, but perhaps $d = 30$ is a good place to start. You can choose the set of d landmarks at random, or use a more sophisticated approach. **Explain your choice of landmarks and write a function to return these landmark nodes**.

4) **Computing Similarities to Landmarks (5 points):** In part 1), you wrote a function to compute feature vectors for each node. Assuming Graph 1 has N_1 nodes and Graph 2 has N_2 nodes, **write a function that computes a similarity measure in this 16-dimensional space between each of the nodes in Graph 1 and Graph 2 to each of the d landmarks.** So, you should en up with a matrix, $\mathbf{C} \in \mathbb{R}^{(N_1+N_2)\times d}$.

5) **Extract Landmark $\times$ Landmark Matrix (5 points):** As you know, the $\mathbf{C}$ that you constructed contains the d landmark nodes! Write a function to construct $\mathbf{W} \in \mathbb{R}^{d \times d}$ submatrix of $\mathbf{C}$ where the similarities between the landmarks were stored.

6) **Embedding via Landmarks (5 points)**: Given Theorem 3.1 in `https://arxiv.org/pdf/1802.06257.pdf`, we can compute the collective node embedding matrix (across Network 1 and Network 2), $\tilde{\mathbf{Y}} \in \mathbb{R}^{(N_1+N_2)\times d}$, as

$$\tilde{\mathbf{Y}} = \mathbf{C}\mathbf{U}\mathbf{\Sigma}^{1/2}$$

.

Recall that here, $\mathbf{U}$ and $\mathbf{\Sigma}$ are obtained through an SVD on the pseudo inverse ($\mathbf{W}^{\text{pinv}}$) of the (landmark $\times$ landmark) similarity matrix, $\mathbf{W} \in \mathbb{R}^{d \times d}$ extracted from $\mathbf{C}$.

$$\mathbf{W}^{\text{pinv}} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$$

Hints: These are useful for pseudoinverse (`https://numpy.org/doc/stable/reference/generated/numpy.linalg.pinv.html`) and SVD (`https://numpy.org/doc/stable/reference/generated/numpy.linalg.svd.html`).

Given this information, write a function to compute $\tilde{\mathbf{Y}}$.

7) **Putting it All Together Visualization 1 (5 points):** You have now defined an embedding for all nodes in Networks 1 and 2 in some d-dimensional space through $\tilde{\mathbf{Y}}$. **Use your favorite dimensionality reduction method of choice to project the collective set of nodes in Networks 1 and 2 into two dimensions. Color the nodes by which network they are from. Comment on any observations.**

8) **Alignment Between Graphs (5 points):** Given $\tilde{\mathbf{Y}}$, **calculate a similarity score (your choice) between each node in Network 1 and every node in Network 2.**

Problem 2 continued on next page...

9) **Creativity (5 points):** Now that you have the entire pipeline in place, play around with it a bit. For example, considering changing how you define the features for nodes in part 1), changing the value of d, changing how you choose landmarks, or anything else that is interesting to you! **Re-run steps 1-7 with your modification and comment on how it changes the interpretation of alignment between Network 1 and Network 2 given in $\tilde{\mathbf{Y}}$.**

10) **Creativity Part 2 (5 points):** Imagine a collaborator dropped these two networks on your desk. They are paying you from their grant, so you need to produce something to give them. **Create a visualization of your choice that reflects something about the similarity between Network 1 and Network 2** (in terms of node alignment, clustering structure, etc).

Congratulations! You implemented REGAL from scratch!