**Cohesive Subgroups Lab**

For this lab we will use three datasets:

**KAPTAIL:**

This is a stacked dataset containing four dichotomous matrices. There are two adjacency matrices each for social ties (indicating the pair had social interaction) and instrumental ties (indicated the pair had work-related interaction). The two pairs of matrices represent two different points in time. The names of the datasets encode the type of tie in the sixth letter, and the time period in the seventh. Thus, the dataset KAPFTS1 is social ties at time 1 and KAPFTI2 is instrumental ties at time 2, etc.

**ZACKAR & ZACHATTR:**

ZACKAR is another stacked dataset, containing a dichotomous adjacency matrix, ZACHE, which represents the simple presence or absence of ties between members of a Karate Club, and ZACHC, which contains valued data counting the number of interactions between actors. ZACHATTR is a rectangular matrix with three columns of attributes for each of the actors from the ZACKAR datasets.

**PV504**

PV504 is a 504-actor network of consultants working for an R&D consulting firm. The data are symmetric and valued and represent the number of days that pair of individuals worked on a project together.

**EXERCISES:**

1. Hierarchical Clustering using UCINET with **ZACKAR**  
   1. This section uses the **ZACHE** dataset (you may have to unpack **ZACKAR** using Data | Unpack to create **ZACHE**) and the **ZACHATTR** attribute dataset. Check to make sure you have both, and let one of the facilitators know if you do not.
   2. Now, run SINGLE\_LINK method Hierarchical Clustering (Tools|Cluster|Hierarchical) on the **ZACHC** matrix (specifying the appropriate kind of data). We are using ZACHC What does the output tell you? Why did you get this result?
   3. Now re-run using the WEIGHTED AVERAGE methods on the same data. Why did you get a different result? Which one is more useful in identifying cohesive subgroups from these data?
   4. [**OPTIONAL ADVANCED]** We used ZACHC because Hierarchical Clustering really works best when there is variation in the data, and particularly a range of distinct values as opposed to 1’s and 0’s on which to run the algorithm. Repeat the Hierarchical Clustering Analysis using ZACHE instead of ZACHC. How do the results differ? Which would you find more useful.

1. Girvan-Newman using NetDraw with **ZACKAR**  
   1. Open the **ZACKAR** stacked dataset in NetDraw. It should open to displaying the relation **ZACHE** but if not, make sure it does.
   2. Now, open the attribute file, **ZACHATTR**, using the folder icon with the A next to it.
   3. Run the Girvan-Newman analysis (Analysis | Subgroups | Girvan-Newman) specifying a minimum of 2 and a maximum of 40 clusters desired. It should automatically color your nodes so that nodes are one of two colors. What it has done behind the scenes is color based on the ngPart\_2 partition (a partition with 2 colors). Click on the color palette icon and pull down on the drop down list to select ngPart\_3 to see how it partitions it next. And then ngPart\_4. How useful are these partitions?
   4. Using the color palette, go back to the ngPart\_2 partition. Now, click on the shape palette icon, and select “Club” from the list. This will shape the nodes according to which club the members went to after the split. How well did the Girvan-Newman algorithm predict the affiliation of the club members?
2. Factions using NetDraw with **ZACKAR**  
     
   Now run Analysis | Subgroups | Factions selecting 2 for the desired number of groups. This time, instead of using the color palette, use the “Nodes” tab in the control area on the right hand side of the screen and scroll down to the last attribute, which should be called “Factions 2” and then click the “Color” checkbox. How does factions compare with the Girvan-Newman algorithm in terms of predicting the affiliations? How could you display the Girvan-Newman results, the Factions result, AND the club attribute all at the same time?
3. Putting it all together  
     
   When you ran Hierarchical Clustering, it created an output file called “Part” which has the hierarchical partitioning. This is an actor by partition matrix which is basically like an attribute file, so the actors are down the rows, and each column is one of the hierarchical clustering solutions. The first partition (labeled ‘1’ in column 1), has the most distributed cluster solution (many clusters, few actors put together, many clusters only have one actor in them). Through the last one (the number of columns varies based on the data input), which will put all the actors into one big cluster. Typically, the second to last cluster will have two clusters, the one before that will have three, etc. (Though some data conditions may affect that.)  
     
   Because of this, you can load Part as an attribute file with the ZACKAR dataset in NetDraw. Now click on the checkbox by the word ‘color’ so that the nodes will be colored by whatever attribute you select in the dropdown list at the top of the control region on the Node Tab. Once the checkbox is selected, switch between several cluster solutions (with numbers 1, 2, … n) to see how the hierarchical clustering grouped nodes. The second to last one should be the solution with two clusters. Select that option and UNCHECK the color checkbox. **So, nodes are colored by the Hierarchical Clustering Solution.**  
     
   Now also **shape the nodes by the ngPart\_2 solution**. Change the **rim size** (Properties | Nodes | Rims | Size | Attribute based) and select the **Factions** Solution and set the minimum and maximum values to 1 and 8, respectively. And, finally, **SIZE the nodes based on the attribute CLUB**. Do this using the Properties | Nodes | Symbols | Size | Attribute based to bring up a dialog box. Select “CLUB” as the attribute and set maximum size to 18 and minimum size to 8.   
     
   At this point, the size tells us where each member actually went when the club split. The shape tells us what Girvan-Newman predicted, color what Hierachical Clustering predicted, and rim weight what Factions predicted.   
     
   To what extend did the three algorithms agree with each other (same shape, color, and rim weight)? How well, in general, did they do (same shape, color, rims weight, and node size)? Remember, these algorithms only know what the patterns of relationships are. Do they predict actual behavior (which club they joined) well?
4. Cliques using UCINET and NetDraw with **KAPFTS2**  
   1. If you have not done so before, unpack the **KAPTAIL** using Data | Unpack. Be sure to eliminate any prefix so your filenames match what I list below.
   2. In UCINET run Network | Subgroups | Cliques on **KAPFTS2** with a minimum size of 3. How many cliques do you get? How many actors are in this network? How useful is this?
   3. Visualize **KAPFTS2** in NetDraw. Does this help us identify clique structures?
   4. What about if we open **CliqueOverlap** (which is an actor-by-actor matrix in which each cell holds the number of different cliques that this pair of actors is in together that was created when we ran Cliques in UCINET). Start increasing the filter at the bottom of the “Rels” tab on the control panel on the right side of the screen up from 1 using the “+” button. Does this indicate there is a significant or minimal overlap between actors in cliques in this network? (Redraw the network when you think you have filtered the data enough.) What does this visualization mean?
   5. Now set the filter back down to 0 and open **CliqueSets** in Netdraw and redraw the picture (lightning bolt). This is a two-mode network where lines indicate actors (typically red circles with names) belong to a specific clique (typically blue squares with numbers). What does this picture convey about the structure of the network? Are there actors who seem embedded in a lot of different cliques? What does that mean about those actors?  
        
      **[OPTIONAL**] Try running Analysis | Centrality on the data inside NetDraw specifying that the data are directed. Then, size the nodes based on OutDegree using the attribute it created and loaded into the nodes tab.
5. K-CORES using NetDraw with **PV504**  
   1. Open **PV504** in NetDraw. Because it is very large, NetDraw does not optimize the layout automatically when opening it. To make the diagram more readable, turn off labels (using the script L button on the icon bar) and arrowheads (the data are symmetric), and then redraw the network. This may take some time, but let it finish. You should begin to see some structure in the network as it draws it.
   2. These are valued data about the number of days individuals worked together on projects. Let’s increase the filtering to be greater than 3 by clicking on the “+” button toward the bottom of the Rels tab in the control region three times. Now redraw the network by clicking on the lightning bolt. Much more structure should be visible.
   3. Now run Analysis | K-Cores. It will automatically color the nodes according to their k-Core. Select the Nodes tab, and pull down to the \*K-core attribute, and use the “s” button below the values to step through the k-cores from 0 to 10. What does this tell you about the network?
   4. Since all nodes of a higher “coreness” are automatically members of the lower cores, we’d like to step down from the highest coreness, to the lowest, but cumulatively. To do this, press the “a” button below the values in the control region to select all the check boxes, then check the “i” button to “inverse” the selection (i.e., uncheck everything that is checked and check everything that is unchecked). This should leave no boxes checked and a blank screen. Now check the box next to the highest value (it should be 10) and look at the graph. Now ALSO check the box next to the second highest value. Repeat until you have checked all boxes. What could you see from stepping “down” the k-cores that was not obvious stepping “up” them?