SNA Basics #7
Centrality & Power

Centrality is one of SNA's oldest concepts. A central actor can be someone who has numerous ties to other actors (degree), someone who is closer (in terms of path distance) to all other actors (closeness), someone who lies on the shortest path (geodesic) between any two actors (betweenness), or someone who has ties to other highly central actors (eigenvector). In some networks, the same actors will score high on all four measures. In others, they won't. There are, of course, more than four measures of centrality. The current version of UCINET includes more than 20, and UCINET does not include them all. ORA and Pajek include several as well. Pajek now includes one (Laplacian centrality), which was developed with dark networks in mind. For this exercise, we will once again use the Anabaptist Leadership data, which are included in the zip file, "`SNA Basics #7(Data)`."

Part I – Centrality and Power in UCINET and NetDraw

Network
>Centrality and Power

1. UCINET allows analysts to compute several different types of centrality. You can find these in the *Centrality and Power* submenu, which is found under the *Network* menu. The most common measure of centrality is degree, which in an undirected, dichotomous network is simply a count of the number of ties that an individual actor has (i.e., the number of the neighbors). In the case of valued data, degree centrality equals the sum of the values of the ties. It also means that any calculations of network centralization and normalized degree centrality may yield scores greater than one. In this case, we're working with a dichotomous network

Network
>Centrality and Power
>_Degree (legacy)

Network
>Centrality and Power
>Degree

2. Select the old *Degree* centrality command found in the *Network>Centrality and Power* submenu, which brings up the degree centrality dialog box (Figure 1). Using the "…" button to the right of the *Input Dataset* drop list, select the `Anabaptist Leaders.##h` data file. Make sure that you indicate that it is a symmetric matrix, accept UCINET's other default settings, and click OK. The output log lists, from highest to lowest, the degree centrality of each actor in terms of raw degree, normalized degree and share (which sums to 1). The output log also provides descriptive statistics such as average degree centrality, standard deviation, minimum, maximum, network centralization and so on. You might want to compare this output with the output of the new degree centrality function. The legacy function presents the results in rank order; the new function does not.

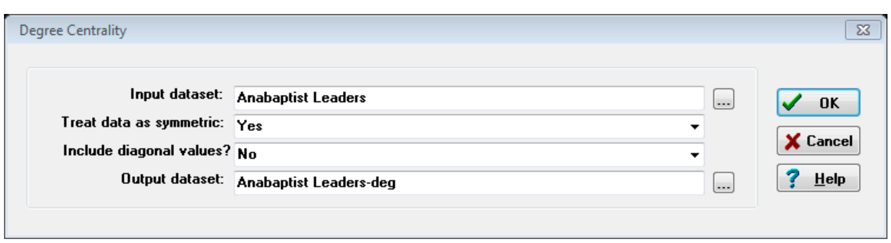


Figure 1: Degree Centrality Dialog Box

3. Closeness centrality assumes that the closer an actor is to all other actors, the easier information may reach it, and as such enjoys higher centrality. The closeness centrality of an actor is based on the total distance between one actor and all other actors, where larger distances yield lower closeness centrality scores. Closeness centrality cannot be calculated when a network includes isolates or is disconnected because you cannot calculate the distance between an isolated actor or a group of actors to other actors in the network. Thus, before calculating closeness centrality in a network, we need to first determine whether it has isolates or is disconnected. This can be done either visually or by running component analysis. If there is more than one weak component, then the graph is disconnected. Also, if you run the standard closeness centrality measure (Freeman), UCINET's output issues a warning that the network is disconnected and thus closeness should not be estimated.

Data>Filter/Extract >Main Component

4. If the network is disconnected, one solution is to extract the largest component, and just estimate closeness centrality with it. If you recall, we can extract the largest component with the *Data>Filter/Extract>Main Component* command. Another solution is to estimate an alternative closeness measure known as is the average (of the sum of) reciprocal distances (ARD). It is attractive because it treats the reciprocal of infinity equal to zero, which means it can be used with disconnected networks. Currently, it only appears to be implemented in UCINET and R (statnet).

Network >Centrality and Power >Closeness (old)

5. Let's estimate closeness centrality of the leadership network using both closeness measures. We'll begin with Freeman closeness with the *Network>Centrality and Power>Closeness (old)* command, which brings up a dialog box similar to Figure 2. Note that the default is Freeman closeness. Click OK and view the results.

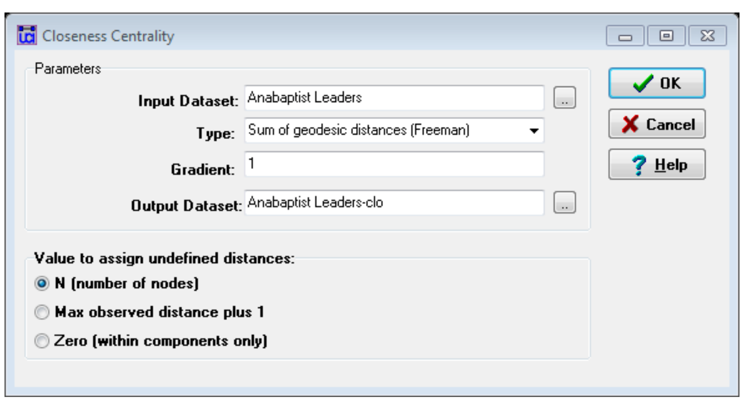


Figure 2: Closeness Centrality Dialog Box (Freeman Closeness)

Network >Centrality and Power >Closeness (old)

6. Now let's estimate ARD closeness centrality. For this, we use the same command but select a different option (Figure 3). Note that average reciprocal distance (ARD) option is called the sum of reciprocal distances (see the dialog box below). Note that there is another option called "Average of Reversed Distances (Valente-Forman)," which can easily be confused with ARD. What we want, though, is the sum of reciprocal distances option, which generates two sets of scores: the sum of reciprocal distances and ARD, the latter being normalized scores. ARD and Freeman closeness tend to correlate quite highly with one another when analyzing networks that are not disconnected. This is good because it suggests that both measures are capturing the same dynamic.

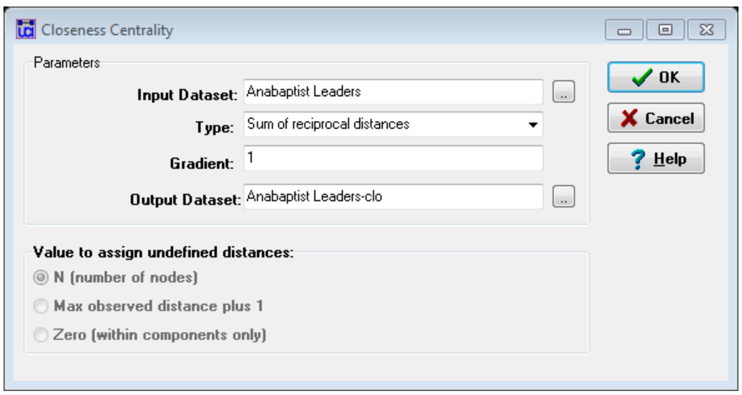


Figure 3: Closeness Centrality Dialog Box (ARD Closeness)

Network >Centrality and Power >Freeman Betweenness >Node Betweenness

7. Betweenness centrality differs from degree centrality in that it assumes that an actor has power over any two other actors when it lies on the shortest path between them in a given network of relations. In order to calculate betweenness centrality for the `Anabaptist Leaders` choose the *Freeman Betweenness> Node Betweenness* command found under the *Network>Centrality and Power* submenu, accept UCINET's defaults, and click OK.

Network >Centrality and Power >Eigenvector centrality

8. Eigenvector centrality assumes that ties to highly central actors are more important than ties to peripheral actors. Thus, it weights an actor's centrality by the centrality scores of its neighbors (i.e., the actors to which it has ties). To compute eigenvector centrality in UCINET, select the *Eigenvector centrality* command under the *Network>Centrality and Power* submenu. In the resulting check the option to force scores to be positive (see Figure 4). UCINET's output first lists a series of eigenvalues before listing the eigenvector centrality scores for each actor. Also, unlike the output for the previous centrality algorithms, the output lists the actors in the order they appear in the dataset, not in rank order.

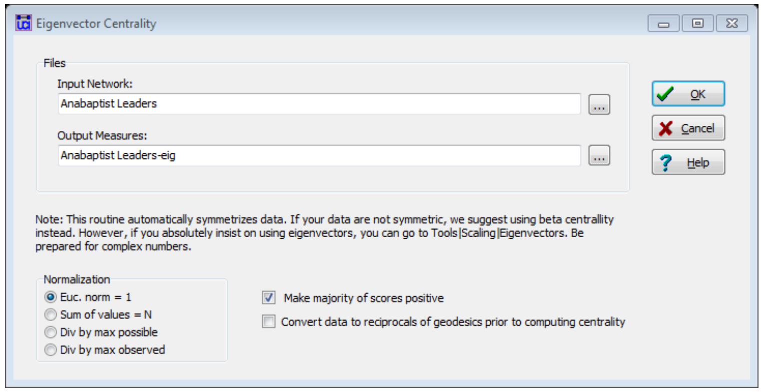


Figure 4: UCINET Eigenvector Centrality Dialog Box

Network >Centrality and Power >Multiple Measures

9. UCINET includes a handy *Multiple Measures* commands under the *Network> Centrality and Power* submenu, which computes numerous centrality scores, including some that we haven't considered. Note that you can get raw or normalized scores, treat the data as valued or dichotomized (binarized), and tell it whether the data are directed or not, or let UCINET detect the direction on its own. However, because it computes multiple scores, it cannot rank the actors by their scores but rather it presents the results in the order the actors appear in the dataset. If you ask UCINET for all of the scores, I recommend selecting the normalized option; otherwise, UCINET returns "farness" scores for Freeman closeness.

10. We can test to see how much the scores correlate with one another using the *Tools>Similarities* command. This brings up a dialog box (Figure 5) where correlations are the default. Choose as your input dataset the one just created in the previous step, make sure that the correlation option is chosen, and indicate that you want to compare similarities among the columns. Click OK, and you'll generate a log output similar to Figure 6 (next page). Note that all of the scores correlate relatively highly with one another although betweenness is something of an outlier. Note also how highly correlated ARD and Freeman closeness are (0.971). As I noted above, this suggests that they are measuring very much the same thing.

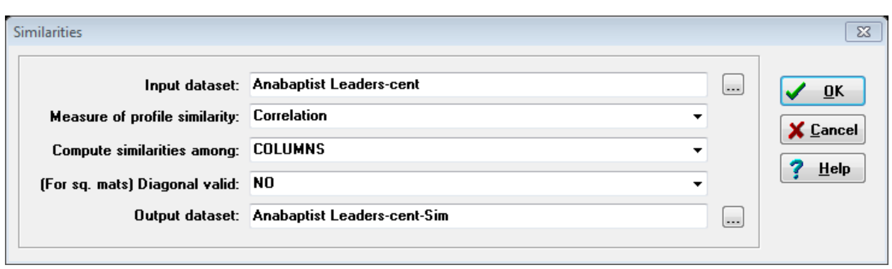
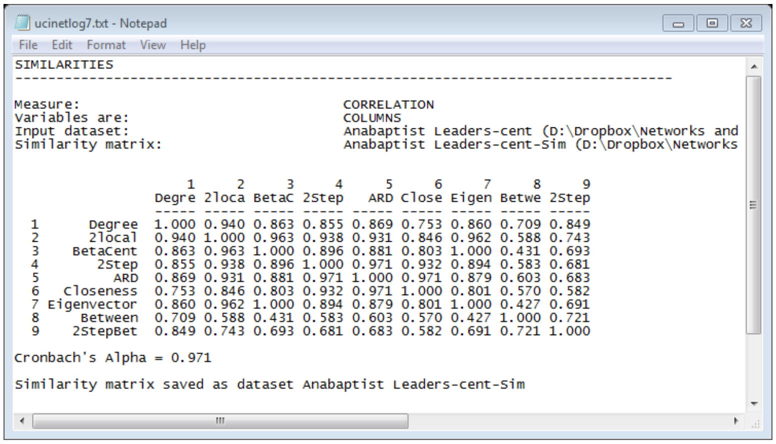


Figure 5: Similarities Dialog Box



```
ucinetlog7.txt - Notepad
File  Edit  Format  View  Help
SIMILARITIES
------------------------------------------------------------------------------

Measure:                                CORRELATION
Variables are:                          COLUMNS
Input dataset:                          Anabaptist Leaders-cent (D:\Dropbox\Networks and
Similarity matrix:                      Anabaptist Leaders-cent-Sim (D:\Dropbox\Networks


                      1     2     3     4     5     6     7     8     9
                  Degre 2loca BetaC 2Step   ARD Close Eigen Betwe 2Step
                  ----- ----- ----- ----- ----- ----- ----- ----- -----
 1        Degree  1.000 0.940 0.863 0.855 0.869 0.753 0.860 0.709 0.849
 2        2local  0.940 1.000 0.963 0.938 0.931 0.846 0.962 0.588 0.743
 3      BetaCent  0.863 0.963 1.000 0.896 0.881 0.803 1.000 0.431 0.693
 4         2Step  0.855 0.938 0.896 1.000 0.971 0.932 0.894 0.583 0.681
 5           ARD  0.869 0.931 0.881 0.971 1.000 0.971 0.879 0.603 0.683
 6     Closeness  0.753 0.846 0.803 0.932 0.971 1.000 0.801 0.570 0.582
 7   Eigenvector  0.860 0.962 1.000 0.894 0.879 0.801 1.000 0.427 0.691
 8       Between  0.709 0.588 0.431 0.583 0.603 0.570 0.427 1.000 0.721
 9      2StepBet  0.849 0.743 0.693 0.681 0.683 0.582 0.691 0.721 1.000

Cronbach's Alpha = 0.971

Similarity matrix saved as dataset Anabaptist Leaders-cent-Sim
```

Figure 6: Correlation of Centrality Scores

[NetDraw]
File>Open
>UCINET dataset>Network

File>Open
>UCINET dataset
>Attribute data

Properties>Nodes
>Symbols>Size
>Attribute-based

11. Centrality measures of individual actors are essentially attributes of actors, which means that we can visualize them like any other ordered attribute. In *NetDraw* open the `Anabaptist Leadership` data file and energize it choosing one of NetDraw's layout algorithms. Next, read in the `Anabaptist Leadership –cent` centrality file that you generated above as an attribute file, using NetDraw's *File>Open>Ucinet dataset>Attribute data* command. Next, using the *Properties >Nodes>Symbol>Size>Attribute-based* command, select various centrality measures in the "Size of Nodes" dialog box (See Figure 7) to vary the size of the nodes. Note that NetDraw sets a minimum size for the nodes; this is helpful because sometimes a node's centrality score equals 0, and if there wasn't a minimum size, the node would disappear.

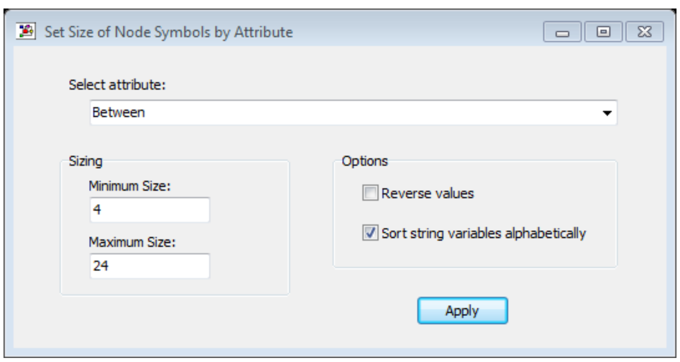


Figure 7: NetDraw's Size of Node dialog box

Analysis
>Centrality measures

Transform
>Node Attribute Editor

12. NetDraw also calculates centrality measures using the *Analysis>Centrality measures* command. This brings up a dialog box where you can choose which centrality measures you want NetDraw to calculate. Click OK and then open the *Node Attribute Editor* to compare calculations made by UCINET and NetDraw. All should be the same except closeness centrality.

Part II – Centrality and Power in Pajek

File>Network>Read

Network>Create Partition >Degree >Input, Output, All

Network>Create Vector >Centrality>Degree >Input, Output, All

1. Read the `Anabaptist Leadership` network file into Pajek. *Degree* can be either a discrete (i.e., raw count of ties) or continuous (i.e., normalized) attribute of an actor, so in *Pajek* it can be calculated as either a partition or vector. To calculate raw degree centrality in *Pajek*, choose either the *Input, Output* or *All* option found under the *Network>Create Partition>Degree* submenu. *Input* counts all incoming ties (indegree), *Output* counts all outgoing ties (outdegree), and *All* counts both. If you are analyzing an undirected network (like this one), it doesn't matter which one you choose. If you want to calculate normalized degree centrality, you can choose either the *Input, Output* or *All* option found under the *Network>Create Vector>Degree* submenu. It would be nice if we could calculate both with just one command, but currently we cannot. Estimate both raw and normalized degree centrality for the Alive Combined Network.

Network >Create New Network >Transform> Arcs→Edges>All

2. As noted above, with an undirected network it makes no difference whether we select *Input*, *Output*, or *All*. Thus, in Pajek we generally want to make sure that we are working with a directed or undirected network. If we aren't, we should ***always*** symmetrize (i.e., make it symmetric) our data ***before*** calculating degree centrality, using the *Network>Create New Network>Transform>Arcs→ Edges>All*. That is, always symmetrize unless, as we do in the next lab (SNA Basics #8), we want to use directed data to estimate measures of prestige.

Partition>Info

3. After estimating both raw and normalized degree centrality, to see which actors are most central, use Pajek's *Partition>Info* command, which calls up a dialog box (Figure 8) that asks whether you want to see the top ranked actors in terms of this vector. If you want the top 10 actors in terms of degree centrality, type a "10" in this box (type a -10 if want the bottom 10). Clicking OK brings up another dialog box (not shown); accept Pajek's default and click OK, which will generate a report similar to Figure 9. As you can see John Hut ranks highest (with a degree centrality score of 16), followed by Wilhelm Reublin, Ulrich Zwingli, and so on. Pajek also provides information on the number (frequency) of actors at each level of degree (e.g., 5 actors have a degree centrality of "1," 10 have a degree centrality of "2," and so on).

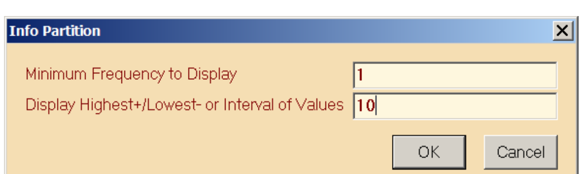
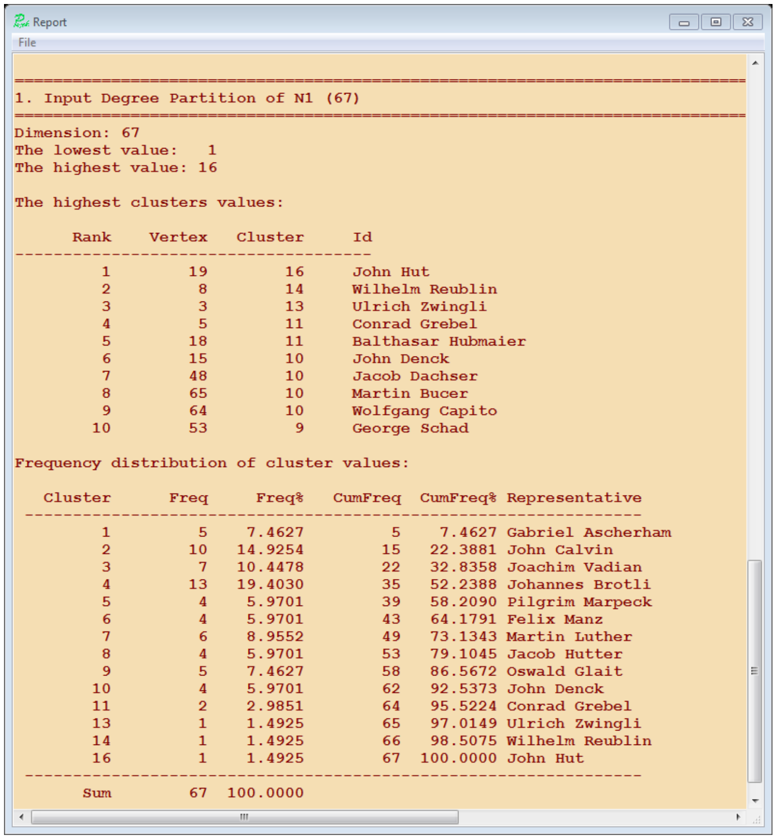


Figure 8: Pajek's Partition Ranking Dialog Box




```
===============================================================================
1. Input Degree Partition of N1 (67)
===============================================================================
Dimension: 67
The lowest value:   1
The highest value: 16

The highest clusters values:

      Rank    Vertex   Cluster      Id
-------------------------------------
         1        19        16      John Hut
         2         8        14      Wilhelm Reublin
         3         3        13      Ulrich Zwingli
         4         5        11      Conrad Grebel
         5        18        11      Balthasar Hubmaier
         6        15        10      John Denck
         7        48        10      Jacob Dachser
         8        65        10      Martin Bucer
         9        64        10      Wolfgang Capito
        10        53         9      George Schad

Frequency distribution of cluster values:

   Cluster      Freq     Freq%   CumFreq  CumFreq% Representative
 ---------------------------------------------------------------
         1         5    7.4627         5    7.4627 Gabriel Ascherham
         2        10   14.9254        15   22.3881 John Calvin
         3         7   10.4478        22   32.8358 Joachim Vadian
         4        13   19.4030        35   52.2388 Johannes Brotli
         5         4    5.9701        39   58.2090 Pilgrim Marpeck
         6         4    5.9701        43   64.1791 Felix Manz
         7         6    8.9552        49   73.1343 Martin Luther
         8         4    5.9701        53   79.1045 Jacob Hutter
         9         5    7.4627        58   86.5672 Oswald Glait
        10         4    5.9701        62   92.5373 John Denck
        11         2    2.9851        64   95.5224 Conrad Grebel
        13         1    1.4925        65   97.0149 Ulrich Zwingli
        14         1    1.4925        66   98.5075 Wilhelm Reublin
        16         1    1.4925        67  100.0000 John Hut
 ---------------------------------------------------------------
       Sum        67  100.0000
```

Figure 9: Pajek Report Window (Raw All Degree Partition)

Draw>Network + First Partition + First Vector

Options>Value of Lines >Similarities

Options>Size>of Vertices

Layers >Type of Layout>2D

Layers>In y direction

4. Next, with the "Anabaptist Leader.net (67)" showing in the top network drop-down menu, the "All Degree partition of N1 (67)" partition showing in the top partition drop-down box, and the "All Degree of N1 (67)" vector highlighted in the top vector drop-down box, view the network with node size adjusted for degree centrality by selecting the *Draw>Network + First Partition + First Vector* command and then using one of the 2D layout algorithms. Be sure that the *Options>Value of Lines>Similarities* command is checked. In addition, if the node sizes do not seem to vary in size, then you may need to adjust the *Size> of Vertices* option under the *Options* menu. Size "8" works well in this case, but sometimes "0" is preferable because it tells Pajek to automatically adjust the size of the nodes. Next, after telling Pajek that this is a 2D layout with the *Layers> Type of Layout>2D* command found in the Draw screen, instruct Pajek to layer the drawing in the "y-direction" using the *Layers>In y direction* command. This should place John Hut at the bottom of the drawing (i.e., the person with the highest degree centrality) and several (i.e., those with the lowest degree centrality) at the top. If you hold down the X key, you can rotate the drawing so that Noordin is at the top. Now your drawing is layered in terms of degree centrality with everyone having the same degree centrality located at the same horizontal level (see Figure 10).

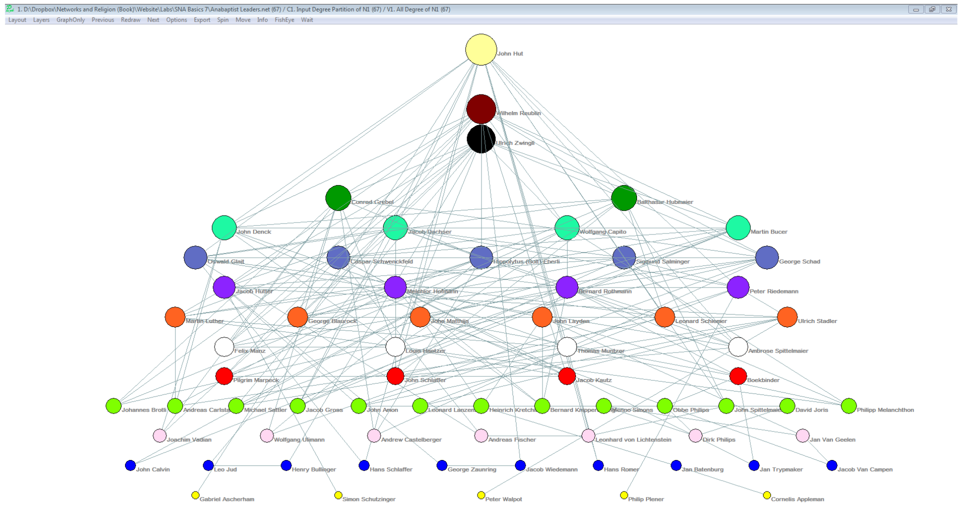


Figure 10: Pajek Drawing of Anabaptist Leader Network, Layered by Degree

Network>Create Vector >Centrality>Closeness >Input, Output, All

Draw>Network + First Vector

5. In Pajek, the computation of closeness, betweenness, and eigenvector centrality is straightforward. Since all are continuous rather than discrete measures, the commands are located in *Network>Create Vector>Centrality* submenu. To compute closeness with undirected networks, you may choose the command *Input*, *Output*, or *All*, all of which yield the same results. If the network is disconnected, Pajek creates a vector of closeness centrality scores where isolates are assigned a score of ‘0’). This creates a vector that you can view with the Noordin network using the *Draw>Network + First Vector* command.

Network>Create Vector >Centrality>Betweenness

Network>Create Vector >Centrality >Hubs-Authorities

Draw>Network + First Vector

6. Betweenness centrality is calculated using the *Network>Create Vector> Centrality>Betweenness* command. Pajek estimates eigenvector centrality with its *Network>Create Vector>Centrality>Hubs-Authorities* command. When analyzing a directed network (as we will in SNA Basics #8), hubs and authority scores differ from eigenvector centrality, but when analyzing an undirected network, the scores are the same. Both the betweenness and hubs-authorities commands create vectors that you can view with the Anabaptist Leader network using the *Draw>Network + First Vector* command.