KliqueFinder © December 14, 2004

User's Guide Ken Frank

The user may find it helpful to refer to "Identifying Cohesive Subgroups" (in *Social Networks*, 1995, Volume 17, pages 27-56) and "Mapping Interactions within and Between Cohesive Subgroups" (in *Social Networks*, 1996, Vol 18, pages 93-119). Both by Kenneth A. Frank.

The Algorithm 1

Running the Program 1

Setups 1

Running KliqueFinder 2
Datafile-- List 3
Datafile-- Matrix 3
A priori placements from another file 4
The 'p' Option 4
The 's' Option 4
Titles 5
Labels 5
Excluding actors from the analysis 5

Options for Running the program 6
Start subgroups (numdyad, startgrp, dydtriad, direct, lookt, thresht,raseed) 6
Attaching actors to subgroups (noattach) 7
Objective Function to be maximized (structeq, network, actrsqr, quantype, netlev, squarit, pergroup, colwt, rowwt). 7
Changes in the Data Matrix (unweight, symmat, transpos) 9
Proximity of actors to subgroups and Inhibiting the assignment of actors to subgroups (nearval, pctile). 9
Boundary Spanning 9
Convergence (stopval, kcount2, quickend) 10
Evaluation of solution (baseval, topval, neval, numres, newgrps) 10

Options for determining printout 11

Interpreting Output 13
COMPACTNESS: A PRIORI GROUPS 14
COMPACTNESS: AFTER ASCENT 21

Making the Plots 24

The basics: All You need to do 26

The Algorithm

Kliquefinder is based on a general algorithm for identifying cliques (clusters or subgroups) of actors in network data. All materials are the property of Kenneth Frank and are not to be reproduced without his permission.

The general algorithm can be described as iterative partitioning, and takes the following steps:

1)Assume that all actors are currently assigned to subgroups
2)Obtain the change in an objective function that would occur if actor i were reassigned to subgroup g for all actors i and subgroups g
3)Make the reassignment that would result in the maximal increase of the objective function.
[The algorithm converges when all actors are in their optimal subgroups]

Because the algorithm is general, and can be applied to many objective functions and their variants, the user is required to specify options to the program.

Running the Program

KliqueFinder requires many parameters (partly because it was built in stages and has not yet been revised. The parametrs are contained in the files below:

hitlist	contains list of actors which should be deleted from the analysis
kliqfind.par	the parameters which guide KliqueFinder in how to cluster. Explanations of the parameters are in this file
labels	contains labels for the subgroups
printo	print options
simulate.par	parameters if you wish KliqueFinder to generate simulated data sets
titles	User specified 3 titles to characterize the data

Setups

In order to copy these files into the directory from which you are going to run KliqueFinder you should type at the Unix prompt:

% setup basic
Once you have run setup basic once, you need never do it again.
Now that you have copied the files into your directory, you may change parameters by editing the

files. But since there are so many parameters and files, I have developed some macros that setup some basic parameter sets. For example, if you would like to run KliqueFinder on a new data set that you have never analyzed before, you can setup all the parameters by typing:
% setup newdata

The other setup that you may have particular use for is:
% setup repeat2
We use "repeat2" when we are going to run KliqueFinder on data in which we have already found subgroups and we just want KliqueFinder to do a smaller set of analyses, such as to plot the people and subgroups in 2 dimensions or to calculate centrality measures.

NOTE: IF YOU WOULD LIKE TO LEARN ABOUT THE OPTIONS AVAILABLE UNDER KLIQUEFINDER AND SET THEIR VALUES INTERACTIVELY, SUBMIT AN EMPTY DATA FILE TO KLIQUEFINDER, OR, TYPE "kliqfind datafile p ". THE LETTER 'p' INDICATES THAT YOU WOULD LIKE TO ENTER THE PARAMETERS INTERACTIVELY. THE PROGRAM WILL GUIDE YOU THROUGH A BRANCHED SERIES OF QUESTIONS AND, BASED ON YOUR RESPONSES, WILL ASSIGN THE APPROPRIATE VALUES FOR EACH OF THE DESIGNATED OPTIONS.

Running KliqueFinder

To actually run kliquefinder (Currently available only for SUN unix) you need to type:
% kliqfind datafile [option] placefile

Here, datafile represents the file which contains the raw data, and [option] can equal:

'l' if the data are in list format
'g' if you would like the assignment of actors to a priori subgroups to be based on a second file, placefile, and
'b' if data are in list format and you would like to use the a priori assignments from a second file.
'p' if you want to set parameters interactively
's' if you want kliquefinder to simulate data sets for you

Note that if you improperly specify a matrix file as a list file or vice versa KliqueFinder will probably figure out eh correct format.

Kliquefinder will look for information in the following files:
datafile, placefile, kliqfind.par, titles, labels, hitlist, simulate.par

Datafile-- List

Raw data representing connections between actors in the network. The data can be either in list format or in matrix format. If the data are in list format '**l**', a typical entry might look like this:

```
          1          3          4      [actual line in data file]
1234567890123456789012345678901      [column indicator, not needed]
```

This would indicate that actor 1 initiates a connection to actor 3 with a value of 4. The second line -- "012345 ..." -- appears here only so that you may see that the format is 3I10. That is, 10 integer spaces are given to each value. The last value may be any integer.

Now, to indicate the a priori subgroup placement of an actor, a typical line might look like this:

```
          1      99999          7      [actual line in data file]
1234567890123456789012345678901      [column indicator]
```

The value of "99999" indicates that this line in the file contains information regarding the a priori subgroup placement of actor 1, and not information about the connection of actor 1 to some other actor. In this case, we see that actor 1 is a priori placed in subgroup 7. All the data regarding actor 1's a priori placement and actor 1's connections to other actors might look like this:

```
          1          3          4
          1      99999          7
          1          5          2
          1          6          3
```

Note that an actor need not have any entries, or it may omit either an a priori placement or connections to other actors. But an actor which does not connect to other actors, nor is the object of other actor's connections, will be removed from the network before analysis.

If you have your data in a list format, but not with the proper spacing, you may convert it to the proper spacing using a simple sas program. First, copy your data into a file called "badform.list" :

% cp mydata.list badform.list

Next, run convert.sas:

% sas convert

Your data are now in a file called "ready1.list" which can be entered into KliqueFinder

Datafile-- Matrix

The same information given for actor 1 above can be represented in matrix format for all actors in the network. To represent the information for actor 1 as given above, we would have:

```
070004023  [actual line in data file]
**0123456  [appears here to indicate column number]
```

The "07" in the first two places indicates the a priori subgroup placement of the actor (note that only 99 a priori subgroups can be used). Beginning in the third column, each entry represents actor 1's

connection to each of the other actors in the network. Therefore there is a value of "4" as the 3rd entry because actor 1 initiates a connection to actor 3 with a value of "4". Note that here the values connecting actors must be integer weights ranging from 0 to 9. Note also that the first line actually appears in the data file while the second line appears here only so that you may see the column numbers of the data.

The matrix format is more restrictive than the list format. It may be, however, easier for you to scan a small data set if it is in matrix format.

A priori placements from another file

Sometimes it may be desirable to indicate a priori subgroup placements from another file. If you should want to do this, you will need to indicate either the option "g" if original data are in matrix format, or the option "b" if original data are in list format, and then indicate the file containing the a priori placements. The format for the file containing a priori placements is:

```
 Intern ID Extern ID  Subgroup
12345678901234567890123456789
```

Note that tyou will not really have to concern yourself with the internal ID's (the need for the two id's is so that Kliqfinder will be compatible with earlier versions of the program). The format above is 3F10.5, that is, each value is a real value with ten spaces, 5 after the decimal. For instance, if we wanted to indicate that actor 1 was in subgroup 6, we would have the following entry:

```
   1.00000   1.00000   6.00000
12345678901234567890123456789
```

This format is identical to the format kliquefinder uses to express the final placements of actors based on the algorithm. Therefore placements can be saved and reentered to circumvent the need to work through all of the iterations for only small changes in the data file or the output.

The 'p' Option

If the user indicates the option 'p', Kliquefinder will prompt for each parameter guiding the clustering process interactively in a branched series of questions.

The 's' Option

One may use Kliquefinder to generate and analyze multiple simulated data sets by indicating the option 's'. In this case, the original data set is <u>not</u> used. The user may specify the number of simulated data sets to analyze, as well as ranges for the number of subgroups to which actors should be a priori assigned, the number of actors per a priori subgroup, the number of connections an actor initiates to other actors, the weights assigned to the connections, and the size of the network. These values are stored in a file named: simulate.par.

Titles

Kliquefinder reads three titles from a file called "titles." The titles may be up to 20 characters long, and are in the following format:

```
              Title 1               Title 2               Title 3
12345678901234567890x12345678901234567890x12345678901234567890
```

Again the numbered row is present just so that the user may easily use the 3A20 format allowed for the titles.

Labels

In addition, the user may assign labels to each of the subgroups in kliquefinder through the file labels, which has the following format:

```
    A          B          C          D          E          F          G
123456xxxx123456xxxx123456xxxx123456xxxx123456xxxx123456xxxx123456
```

Here, generic alphanumeric labels are used, although the user may specify each label up to 6 characters. The user may specify up to 250 labels.

Excluding actors from the analysis

A user may exclude certain actors from the analysis by editing the contents of the file called "**hitlist**." The file has the following format:

```
Num ID1 ID2 ID3
 123x123x123x123
```

The format of the file is thus (I3,1X). The first entry indicates how many actors you wish to delete. Each subsequent entry indicates the ID# of an actor you wish to delete. For example: 003 009 007 012

Would delete three actors (7,9,12) from your analysis.

You may list up to 250 actors to delete from your analysis.

All one truly needs to run KliqueFinder is a properly formatted data file. The following options, specified in the file "kliqfind.par" indicate how Kliquefinder should proceed in finding and evaluating a placement of each actor into cliques. The final section includes a listing of print options contained in the file "printo" which kliquefinder uses to determine what type of output the user would like.

Options for Running the program

Options for running the algorithm along with complete explanations are contained in the file "kliqfind.par" . The user may use any unix text editor to modify the values in this file. The user may also set the parameters interactively by submitting an empty data file to kliquefinder or by using the letter 'p' as the option after giving the name of the data file.

Below is a description of each of the parameters (brief but analogous descriptions are given in the file itself).

Start subgroups (**numdyad, startgrp, dydtriad, direct, lookt, thresht,raseed**)

Kliquefinder needs some way to initiate subgroups which can be specified by the user. First, the user must specify how many subgroups the program should start with. The value for this parameter is placed in **numdyad**, and may range from 0 to 251. Note that if the value of 0 is used, the algorithm treats each actor as a unique subgroup to begin this algorithm. This may be a perfectly sensible and functional value for this option.

Next the user must determine how the subgroups should be initiated. There are three options:

1 if you want to use similarities obtained by matrix multiplication to identify seeds for subgroups.

2 if you want a random start (with numdyad subgroups)

3 if you want to start from a priori subgroups

The user assigns a value of 1,2, or 3 to the variable **startgrp** according to her preference from the above list.

Matrix multiplication (**Direct, lookt, thresht)**

The matrix multiplication used is comparable to that long used in sociometric analysis (see Harary and Ross, 1956) to find to cliques of actors. In the application here, each association between any pair of actors is assigned a standardized value with mean and expectation derived from the hypergeometric distribution where the size of target subgroup is set equal to 1 (That is, n_g=1, as referred to in "Iterative Partitioning Cluster Algorithms for Measures of Connectivity on Sociometric Data", by Kenneth Frank, 1991). The standardized values are then converted to values ranging between -1 and 1 via the inverse Fisher Z transformation. The transformation is effected to ensure that indirect (multiplied) connections will be down-valued.

Through matrix multiplication high values are associated with triples (if **dydtriad**=1) or pairs (if **dydtriad**=2) of actors, indicating that these actors are connected to each other as well as similarly connected to actors throughout the network. The implication is that a pair or triple with a high value will be connected to each other as well as be "attractive" to a subset of actors in the network who may later join this subgroup.

Since matrix multiplication is computationally intensive, the user has the option to limit the number of pairs or triples to be looked at by specifying the value of **lookt**.

Further, the user may specify that a value associated with a pair or triple must be at least

above a threshold to be considered as a candidate for a seed subgroup. This value is specified by **thresht**.

The user has the option to specify the weight associated with direct connections by assigning a value to **direct**. Thus, if the user wishes to form seed subgroups based primarily on connections that exist within the subgroup, without concern for the subgroup's ability to attract future actors, one might assign a very high value to direct (i.e. 999).

The program will begin with as many non-overlapping seed subgroups as the user specifies in **numdyad.**

Random Assignment (**raseed**)

If the user specifies "random assignment,", actors will be randomly assigned to the number of subgroups indicated by **numdyad**, based on the seed value indicated in **raseed**.

A priori subgroups

If the user specifies a priori subgroups, all actors assigned to a priori subgroups will be placed in those groups.

Attaching actors to subgroups (**noattach**)

By default, Kliquefinder will "attach" all isolated actors to existing subgroups before engaging in the full algorithm. If you do not wish this to occur, assign a value of 1 to **noattach**. The algorithm will then proceed by reassigning actors to subgroups (including isolates), based on maximization of the objective function.

Objective Function to be maximized (structeq, network, actrsqr, quantype, netlev, squarit, pergroup, colwt, rowwt).

Iterative partitioning can be applied to many different objective functions. KliqueFinder can currently maximize objective functions which cover two broad classes. The first identifies subgroups based on a high degree of connectivity within subgroups and a low degree of connectivity between subgroups. This is called maximization based on **connectivity**. The second approach assigns actors to subgroups based on a similarity of their connections to all actors in the network. The is referred to as structural equivalence. An excellent description of maximization based on structural equivalence is given in Panning (1982). The default for KliqueFinder is connectivity. If you would like to maximize based on structural equivalence, assign a value of 1 to **structeq**. In either case, one can relate the maximization of a function to a statistical test of the hypothesis that subgroups are salient versus the hypothesis that subgroups are not salient (See "Iterative Partitioning Cluster Algorithms for Measures of Connectivity on Sociometric Data", by Kenneth Frank, 1991).

If you do choose to maximize structural equivalence, the algorithm has two options. You may either reassign actors to subgroups so to maximize the measure of R^2 (as described by Panning, 1982, where R^2 refers to the R^2 measure of fit that would be obtained if one assigned the designated subgroups to the outcome variables corresponding to the columns of the data matrix), or you may

reassign actors based on Euclidean distances between an actor and the centroid of each currently existing subgroup. The default is to reassign based on Euclidean distances. By assigning a value of 1 to **actrsqr** you indicate that you want to maximize the value of R^2. Because of properties related to the technique of "updating a sum of squares" (Thisted, 1988, pp. 50-51), the value associated with the distance between an actor and a subgroup for maximizing the $R^2 = n/(n+1)$ * euclidean distance from mean. Thus, for large sized subgroups, or instances where the subgroups are all of a fixed size, there is little difference in the effect of the use of the actual R-square versus the Euclidean distance.

If you are maximizing based on structural equivalence, you may or may not have network data. "Network data" is defined as data where the rows and columns of your input matrix consist of identical elements. If you do have network data, it may well be advantageous to consider the connection of an actor to itself to be equal to the largest positive value in the entire matrix. This indicates the actor is highly connected to itself and is thus similar to actors which initiate connections to it. If the user wishes this option, she should assign a 1 to the variable **network**. Otherwise, the diagonal elements of the original data matrix will be as input.

Note: The other options listed under this heading do not apply if you specify structeq=1.

If you choose to maximize connectivity, there are several options for choosing an objective function to maximize. Essentially, all can be derived by obtaining expectations and variances for the level of connectivity between each actor and a given subgroup (under the null hypothesis that subgroups are not salient) and using these values to standardize the actual level of connectivity between an actor and a subgroup (See "Iterative Partitioning Cluster Algorithms for Measures of Connectivity on Sociometric Data", by Kenneth Frank, 1991).

There are three general objective functions (which measure connectivity) which Kliquefinder can maximize. First, if one wishes to maximize Hubert's (1987) measure of compactness (and related measures), one should specify **quantype**=1. This will maximize based on (Observed - expected)/sqrt(variance).

If the user wishes to standardize based on (observed-expected)/sqrt(expected), one should indicate that **quantype**=2. If one wants to maximize the statistic that is actually used in the Pearson chi-square, one must specify, in addition, **squareit**=1, which indicates that the values should be squared (note that the original sign if the unsquared distance is preserved).

Asymptotically equivalent to Pearson's Chi-square would be the measure L, which comes from the likelihood ratio statistics used to test the two models described above. The contribution of each subgroup to the likelihood ratio statistics is (observed)*log(observed)/expected. If one wishes to reassign actors to maximize the measure L, one would assign **quantype**=3.
$\mathbf{2}_1$ is maximized by choosing **quantype**=5 (This is the deault in setup newdata)

There are several options that allow the user to determine at which level the standardization should be determined. The default is to determine a standardized score for each subgroup, and to make the reassignment that would effect the greatest increase in the sum of these subgroup scores. If one wishes to make the reassignment based on a change in score as computed at the network level one would assign **netlev**=1.

By assigning **colwt**=0, one is essentially indicating that an actor should be reassigned based

only on the connections that it initiates to other actors in the network. This implies that the standardization is occurring at the individual level. Therefore **colwt** can be used to determine the level at which the function is evaluated in assigning actors to subgroup. It's complement, **rowwt**, may also be specified. The two should sum to 1, but it is not necessary.

With quantype equal to 1, 2 or 3, one has the option to square the change in objective function (preserving the original sign), by assigning **squarit**=1. This may be especially important when quantype=2, indicating use of Pearson's chi-squared statistic.

The user also has the option to divide the change in a subgroup level statistic by the current size of that subgroup by assigning **pergroup**=1. Thus, an actor would show a positive association to a subgroup only if it were more highly associated with that subgroup than were the actors currently assigned to that subgroup (a stringent criterion indeed!).

Changes in the Data Matrix (**unweight, symmat, transpos**)

Note that if one wishes to treat weighted data as unweighted for the purposes of using either Pearson's measure or the measure L, one may assign a value of 1 to the variable **unweight**. All values in the original data which were greater than 1 will become 1, and all values which were 0 will remain at 0.

The user may also consider all connections between pairs of actors to be symmetric by assigning **symmat**=1. Kliqfinder would assign the value to the connection of actor A and actor B as the connection that actor A initiates to actor B plus the connection that actor B initiates to actor A.

The user may also take the transpose of the initial matrix by assigning **transpos**=1.

Proximity of actors to subgroups and Inhibiting the assignment of actors to subgroups (**nearval, pctile**).

At any given phase of the algorithm, the user may wish to indicate that an actor should not be reassigned to a subgroup unless the actor has a standardized association with that subgroup of a value greater than a given threshold. That threshold is assigned to the variable **nearval**. For instance, one might say that a standardize association (as measured by Hubert's compactness) between an individual and a subgroup must be greater than a value of 1.00, by assigning nearval=1.00 . Because it may be difficult to prespecify the value, the user may indicate that the value of association must be at least as great as the x^{th} percentile of the associations of actors (which are currently assigned to subgroups of size greater than 1) to their subgroups. The value of x is indicated by the variable **pctile**. In practice, the final results have shown to be relatively insensitive to the specification of **pctile**. Nonetheless, the value of .20 has worked well in practice.

The user may use a fixed value of nearval for all iterations by specifying pctile to be greater than 1.0 .

Note: If the user specifies to maximize the objective function at the network level, that is, **netlev**=1, then nearval=.000 and pctile=2.000. This is to ensure that each move at each iteration increases the value of the objective function.

Boundary Spanning

Once a final solution has been identified, output indicating the relationship between each actor and each subgroup may be requested and placed in filename.bound. In addition, a certain screened portion of these may be optionally output in filename.bound2. The value for this screening is **boundval**, which may be thought of similar to nearval. Only information regarding associations higher than boundval will be given in filename.bound2.

Convergence (**stopval, kcount2, quickend**)

Convergence of the iterations may be determined when the reassignment of an actor is less than **stopval**. Typically, a value of .0005 works well for stopval, indicating that the algorithm should run until all actors have been assigned to their optimal subgroups.

The user may also place a cap on the number of iterations employed by assigning an integer to **kcount2.**

The user may circumvent all iterations, obtaining output essentially based on the initial assignment of actors to subgroups by assigning **quickend**=1. This may be advantageous if the user is interested in output based only on the a priori subgroups.

Evaluation of solution (**baseval, topval, neval, numres, newgrps)**

Kliquefinder provides for a monte carlo-like evaluation of the worthiness of a given clustering solution. This consists of drawing samples of solutions representing a range of similarity to the solution produced by the algorithm, and plotting similarity with the solution produced by the algorithm against the value of the objective function associated with each sample solution. A highly linear plot indicates that most good solutions are similar to the one produced by the algorithm, thereby giving the researcher more confidence in the final solution.

The range of similarities is determined by the percent of actors who should change subgroup assignments. This range is specified by **baseval** and **topval**. Note that one can achieve a classic monte carlo test of a given grouping scheme by assigning baseval=0 and topval=0.

The number of samples to be drawn may range from 0 to 9999, and is specified by **neval**.

In the reassignment of actors to subgroups, one can allow for new subgroups to randomly emerge (of size 3) by specifying **newgrps** = 1. Otherwise, actors can only be reassigned to existing subgroups.

Finally, one can search for a second (and in the future perhaps multiple) solutions by indicating **numres** = 2. This tells Kliqfinder to take the largest positive residual from a regression of objective function on similarity to final solution, and use this as the start subgroups for a new solution. This will obtain a good solution that has relatively little in common with the first found.

Options for determining printout

A file named "printo" can be edited to control the printout that the user would like. Below are a list of the options:
The printo file looks like this:

```
4301000000001101000000101101001111110321101 01
  1234567890123456789012345678901234567890123
  0         1          2          3          4
```

first two digits remian fixed -- the number of variables to be read in

value of 0 suppresses the output:
1 causes the output to print

printo output (all output in "clusters" unless otherwise specified)

1 info for each actor about association with current and best group at each iteration.
2 Matrix of choices -- according to group structure
3 matrix of choices -- Actor Order
4 hubert matrix -- 1 if in same group, actor order
5 print out compactness info after initializing groups
6 calculate and print actor X group probabilities in files grpprob.out and grpprob.dat
7 calculate and print actor X actor probabilities in files persprob.out and persprob.dat
8 print out all the "HAVE" and "WANT" moves during iterations
9 calculate and print final groupings for logit analysis in "logit.dat"
10 print the final solution in matrix form (like the input matrix) into "clus.outmat"
11 info on raw boundary spanners (in "boundary")
12 abbreviated boundary spanners -- greater than boundval (in "boundary 2")
13 output and commmands for MDS scaling of requested Group Associations matrices
14 output indicating the group placement of each actor in "placements" (format is 3(f10.5): cluster #, id # as input, group identification)
15 resids and chi-square from logit fit of the effect of actors being in the same group on the probability of a connection.
16 Print ucinet matrix for whole network (in ucwn.DAT)
17 Print ucinet matrix for group associations in ucgc.DAT
18 Print ucinet matrix for within group connections in (ucgrpxxx.DAT)
19 Print ucinet vector of group assignments (for Hubert QA)
20 Table of group blockmodel for structural equivalence
21 Print out compactness Table and matrices for a priori groups (1=yes, 0=no)
22 Print initial info about klique finder and printo tables
23 Print progress about the objective function at each iteration (to the screen, to ?.clusters, and to obfun.dat)
24 operate in interactive mode, indicating output files and advancement of the function

to the screen
25 print out final placements and data in list format
26 give Blau's Measures of groups distance in boundaryb and bound2b
27 add measures of overall compactness and subgroup compactnesses to end of compmeas.dat and compgrp.dat
28 measures of centrality
29 measures of betweenness
30 Group Associations, Compactness
31 Group Associations, Sum of Zi
32 Group Associations, Mean Within Group Connection
33 Group Associations, Proportion Connections in Group
34 Write compgrp stuff to files
35 Actor by Actor listing of weights between actors based on group associations in file called "perxper.?dat"
36 matrix for creating order for groups, 0 = no order, 1=compactness, 2=zi, 3=density, 4=proportion within
37 method for comparing matrices to create an order for groups and actors within subgroups. 1= difference, 2=log of ratio, make the value negative to maximize the upper triangular matrix
38 =1 if you want to kliqfinder to use polar coordinates to place groups and actors within groups in 2 dimensions
39 =1 if actors within groups should be arranged using polar coordinates -- same options apply
40 =1 if you want to resort original ID's (because they're greater than max)
41 =1 if you want to rotate members within subgroups to minimize connections cutting through groups
42 =1 if you want extra info at end of group choices matrix
43 =1 if you want distances between people

The file printo lists each option and provides a single line at the beginning of the file with a string of 1's and 0's. The user may edit this string, with a 1 indicating that the user would like the particular option, a 0 indicating the user would not like this option. NOTE: the program does read all of the lines of the file. This file should not be modified other than editing the first line.

Interpreting Output

KliqueFinder puts most of the output in a file called: mylist.clusters. That is, if you click:

[Run analysis] (or in unix: kliqfind stanne.ilist)

The output will be in stanne.clusters. NOTE THAT KLIQUEFINDER WILL ONLY USE THE FIRST SIX LETTERS OF YOUR FILENAME

Clusters Output. Below is some annotated sample output.

```
                DATE
Wed Nov  2 16:43:09
        INPUT FILE:
    stanne.ilist
           DATA TYPE
                  1
   DATA IN LIST FORM
 USING PRIOR GROUPS:
    stanne.ilist

The objective function to be maximized is:

2
 1

The distances will be taken at
the network level.


Removing actor   18 because the actor  connected to no others in
the network.

KliqueFinder will remove various actors depending on the value
specified for tagalong.

-------------------------------------------

The first set of KliqueFinder output is for your a priori
subgroups (perhaps the departments of the teachers)

                        ASCENT

THE OBJECTIVE FUNCTION AT ITERATION    0 IS    9.81732
    0 ITERATIONS FOR THIS PHASE

           ASKED FOR QUICK END

For each actor, the following table will give the actorID, the
value of the objective function given the actor is in it's
subgroup, the value of the objective function if the actor were
```

to be moved to a better subgroup, and the better subgroup. This is mostly for diagnostic purposes.

```
 GROUP    ACTORS
 (N)
   1             4           7          15          24
(   4)    (0.9817E+01)(0.9817E+01)(0.9817E+01)(0.9817E+01)
BEST G    (0.9817E+01)(0.9817E+01)(0.9817E+01)(0.9817E+01)
 GROUP    (          1)(          1)(          1)(          1)
```

What follows is a table of exchanges by groups for the a priori subgroups

COMPACTNESS: A PRIORI GROUPS

TITLE	NETWORK	OTHER INFO
New School	Professional Discuss	After 11-9-92

Subgroups have been reordered according to the following scheme:

GROUP *(in case the groups were reordered according to centrality)*

OLD	1	2	3	4
NEW	1	2	3	4

CHOICES BY GROUPS
Each Cell Represents The Number of Transactions (Also Known as Connections or Exchanges) Between Actor i (Row) and Actor j (Column)

N		Group And Actor Id				
24		AAAA	BBBBBB	CCCCCCCC	DDDDDD	
		2 1	221 1	11 2	111122	
Group	ID	4475	612214	98133560	769037	
1 A	4	A11.	.....	.....	.....	
1 A	24	1A1.	.....	.1.....	.....	
1 A	7	11A1	.....	.....	...1..	
1 A	15	111A	.....	.....	.....	
2 B	26	.1..	B111..	.....	.....	
2 B	21	.1..	1B....	...1....	....1.	
2 B	12	....	1.B...	.....	.....	
2 B	2	....	11.B..	.....	...1..	
2 B	1	1...	1..1B.	.....	..1.1.	
2 B	14	....	....1B	.....	.....	
3 C	9	....	.....	C...1.11	..1...	
3 C	8	.1..	..1...	.C.1..1.	1.....	
3 C	11	....	.....	11C.1.1.	.1....	
3 C	13	.1..	.1....	111C....	.....	
3 C	3	..1.	.1....	1.11C...	.....	
3 C	5	.1..	1	1.1.1C..	.....	
3 C	6	....	.....	111..1C1	.....	
3 C	20	....	.....	1..1.11C	.....	
4 D	17	.1..	.....	.1.....	D1....	
4 D	16	....	.....	1..1...1	1D1...	
4 D	19	....	.....	1.1.....	11D...	
4 D	10	1...	...1..	.....	...D1.	
4 D	23	....	.1....	.....	.111D.	
4 D	27	.1..	.1....	.....	..1.1D	

Info about marginals is available. See Printo(42)

N	N MADE CONN	NON-ISOLATES	POSS CONNECT
24	24	24	552.00000

UNWEIGHTED CONNECTIONS	% CONNECT
91.00000	0.16486

WEIGHTED CONNECTIONS	AVG CONNECT
91.00000	0.16486

COMPACTNESS

HUBERT'S

# IN-GROUP CON	EXPECTED	STD	STANDARDIZED	P-VALUE
58.00000	21.10145	4.90338	7.52513	0.00000

/ACTOR	/NON ISOLATE
0.31355	0.31355

KLIQUE FINDER'S

# IN-GROUP CON	EXPECTED	STD	STANDARDIZED	P-VALUE
58.00000	22.04348	3.73490	9.62717	0.00000

/ACTOR	/NON ISOLATE
0.40113	0.40113

PREDICTED THETA ($2_{1\ base}$) BASED ON SIMULATIONS

0.76985

CORRESPONDING ODDS RATIO

4.66315

Approximate LRT BASED ON PREDICTED THETA (1 base)

36.06516

ESTIMATE OF THETA 1 subgroup processes = 2_1-$2_{1\ base}$=log(9.81732)/2-.76985

0.37223

From Section 5 of Identifying Cohesive Subgroups

THETA1 IS TAKEN FROM THE FOLLOWING MODEL:
LOG P(Xii'=xii')=theta0+theta1(samegroup)

THETA1 ALSO CAN BE INTERPRETED AS THE
LOG-ODDS OF THE FOLLOWING TABLE:

		CONNECTION NO	CONNECTION YES
IN SAME GROUP	NO	ALPHA	BETA
	YES	GAMMA	DELTA

MEAN WITHIN GROUP CONNECTION (DENSITY)

THE LARGEST NUMBER OF CONNECTIONS INITIATED BY ANY ACTOR IS 5.0000
WHICH IS GREATER THAN OR EQUAL TO THE PRESPECIFIED MAXIMUM OF 1.0000 .
THE FORMER WILL BE USED IN CALCULATING DENSITY MEASURES.

DIRECT ASSOCIATIONS

GROUP	1	2	3	4
LABEL	A	B	C	D
N	4	6	8	6
GROUP				
1	0.83	0.00	0.05	0.05
2	0.12	0.33	0.03	0.13
3	0.12	0.10	0.68	0.08
4	0.12	0.10	0.20	0.37

DIAGONAL ELEMENTS

SUM	/ # GROUPS	/ACTOR	/NON ISOLATE
2.20833	0.55208	0.09201	0.09201

DIAGONAL PER ACTOR

1	2	3	4
0.21	0.06	0.08	0.06

UNIVARIATE STATISTICS ON DIAGONALS

MEAN	VAR	STD	SKEW
0.55208	0.05881	0.24252	0.33944

OFF-DIAGONAL ELEMENTS

MEAN, IN-CENTRALITY (FREEMAN'S IN-DEGREE)
TAKEN AT SUBGROUP LEVEL
0.12 0.07 0.09 0.09
MEAN PER ACTOR
0.03 0.01 0.01 0.01

MEAN, OUT-CENTRALITY (FREEMAN'S OUT-DEGREE)
TAKEN AT SUBGROUP LEVEL
0.03 0.10 0.10 0.14
MEAN PER ACTOR
0.01 0.02 0.01 0.02
RATIO IN/OUT CENTRALITY
3.75 0.69 0.94 0.61
RATIO IN/OUT CENTRALITY PER ACTOR
0.94 0.11 0.12 0.10

UNIVARIATE STATISTICS OFF-DIAGONALS
IN-CENTRALITY

MEAN	VAR	STD	SKEW
0.09306	0.00059	0.02427	0.64776

UNIVARIATE STATISTICS OFF-DIAGONALS
OUT-CENTRALITY

MEAN	VAR	STD	SKEW
0.09306	0.00200	0.04470	-0.72933

TOTAL RELATIONS (FREEMAN'S CLOSENESS)

GROUP	1	2	3	4
LABEL	A	B	C	D
N	4	6	8	6
GROUP				
2	0.13			
3	0.13	0.13		
4	0.13	0.12	0.22	

OFF-DIAGONAL ELEMENTS
MEAN, IN CENTRALITY (FREEMAN'S IN-DEGREE)
TAKEN AT SUBGROUP LEVEL
0.13 0.13 0.16 0.16
MEAN PER ACTOR
0.03 0.02 0.02 0.03

UNIVARIATE STATISTICS OFF-DIAGONALS
IN-CENTRALITY

MEAN	VAR	STD	SKEW
0.14408	0.00031	0.01748	-0.07312

BETWEENNESS / TOTAL RELATIONS (FREEMAN'S BETWEENNESS)

GROUP	1	2	3	4
LABEL	A	B	C	D
N	4	6	8	6
GROUP				
1	0.00	0.25	0.19	0.20
2	0.00	0.00	0.04	0.04
3	0.09	0.07	0.00	0.13
4	0.08	0.13	0.11	0.00

OFF-DIAGONAL ELEMENTS

MEAN, IN CENTRALITY (FREEMAN'S IN-DEGREE)
TAKEN AT SUBGROUP LEVEL

0.06	0.15	0.12	0.12

MEAN PER ACTOR

0.01	0.02	0.01	0.02

MEAN, OUT CENTRALITY (FREEMAN'S OUT-DEGREE)
TAKEN AT SUBGROUP LEVEL

0.21	0.03	0.09	0.11

MEAN PER ACTOR

0.05	0.00	0.01	0.02

RATIO IN/OUT CENTRALITY

0.26	5.01	1.24	1.16

RATIO IN/OUT CENTRALITY PER ACTOR

0.06	0.83	0.15	0.19

UNIVARIATE STATISTICS OFF-DIAGONALS
IN-CENTRALITY

MEAN	VAR	STD	SKEW
0.11054	0.00157	0.03968	-1.20234

UNIVARIATE STATISTICS OFF-DIAGONALS
OUT-CENTRALITY

MEAN	VAR	STD	SKEW
0.11054	0.00581	0.07623	0.83658

NETWORK WIDE MEAN WITHIN GROUP CONNECTION

0.51786

THE ALGORITHM WILL NOW INITIATE IT'S ASCENT TO FIND NEW GROUPS.

```
  STARTING    GROUPS
     TRIAD   ACTOR 1   ACTOR 2   ACTOR 3
         1        15         7         4

STARTING WITH     BEST TRIADS

-------------------------------------------
                         ASCENT

THE OBJECTIVE FUNCTION AT ITERATION    0 IS   26.74419
THE OBJECTIVE FUNCTION AT ITERATION    1 IS   28.33333
THE OBJECTIVE FUNCTION AT ITERATION    2 IS   30.13158
                                       .
THE OBJECTIVE FUNCTION AT ITERATION   14 IS   12.39419
   14 ITERATIONS FOR THIS PHASE

  NO ISOLATES, PHASES COMPLETE
```

THE OUTPUT GIVEN FOR THE A PRIORI SUBGROUPS IS NOW REPEATED FOR THE SUBGROUPS IDENTIFIED BY KLIQUEFINDER

COMPACTNESS: AFTER ASCENT

TITLE	NETWORK	OTHER INFO
New School	Professional Discuss	After 11-9-92

Subgroups have been reordered according to the following scheme:

GROUP					
OLD	1	2	3	4	5
NEW	1	5	3	4	2

CHOICES BY GROUPS
Each Cell Represents The Number of Transactions (Also Known as Connections or Exchanges) Between Actor i (Row) and Actor j (Column)

N		Group And Actor Id				
24		AAAA	BBBBB	CCCC	DDDDD	EEEEEE
		2 1	11 11	2212	2 11	1 2
Group	ID	4475	37869	1307	26124	931560
1 A	4	A11.	.....	....	.....	.....
1 A	24	1A1.	..1..	....	.....	.....
1 A	7	11A1	.....	..1.	.....	.....
1 A	15	111A	.....	....	.....	.....
2 B	13	.1..	B.1..	1...	.....	1.1...
2 B	17	.1..	.B11.	....	.....	.....
2 B	8	.1..	11B..	....	...1.	....1.
2 B	16	....	11.B1	....	.....	1....1
2 B	19	....	.1.1B	....	.....	1.1...
3 C	21	.1..	1....	C1..	.1...	.....
3 C	23	....	...11	1C1.	.....	.....
3 C	10	1...	.....	.1C.	1....	.....
3 C	27	.1..	1	11.C	.....	.....
4 D	2	....	.....	1.1.	D1...	.....
4 D	26	.1..	.....	1...	1D.1.	.....
4 D	1	1...	1	.1..	11D..	.....
4 D	12	....	.....	....	.1.D.	.....
4 D	14	....	.....	....	..1.D	.....
5 E	9	....	1	....	.....	E1..11
5 E	3	..1.	1....	1...	.....	1E1...
5 E	11	....	..11.	....	.....	11E.1.
5 E	5	.1..	.....	....	1	111E..
5 E	6	....	..1..	....	.....	1.11E1
5 E	20	....	1....	....	.....	1..11E

VALUE OF THETA1 FOR THESE SUBGROUPS IS

1.14207

THETA1 IS TAKEN FROM THE FOLLOWING MODEL:
LOG P(Xii'=xii')=theta0+theta1(samegroup)

THETA1 ALSO CAN BE INTERPRETED AS HALF THE
LOG-ODDS OF THE FOLLOWING TABLE:

		CONNECTION NO	CONNECTION YES
IN SAME GROUP	NO	ALPHA	BETA
	YES	GAMMA	DELTA

PREDICTED THETA (1 base) BASED ON SIMULATIONS

0.76985

ESTIMATE OF THETA (1 subgroup processes)

0.37223

THE TOTAL THETA1 IS:

1.14207

Approximate LRT BASED ON PREDICTED THETA (1 base)

8.96081
OBSERVED VALUES FOR CELLS A,B,C, AND D
391.00000 33.00000 70.00000 58.00000
ODDS RATIO, LOG ODDS, (LOG ODDS/2)
9.81732 2.28415 1.14207

PREDICTED VALUES FOR CELLS A,B,C, AND D
378.77707 45.22294 82.22294 45.77706
ODDS RATIO, LOG ODDS, (LOG ODDS/2)
4.66315 1.53969 0.76985

COMPARE TO CHI-SQUARE ON 1 DF
P-VALUE (LESS THAN OR EQUAL TO):
0.00500

This is roughly a test between the models
LOG P(Xii'=xii')=theta0+theta1base(samegroup)
LOG P(Xii'=xii')=theta0+theta1base(samegroup) +
 theta1 subgroup processes(samegroup)

Note that the estimate of 2_1 based on simulated data will not be accurate if your network size or density is outside of the range of the simulated values in Frank (1995). In this case you must generate a unique monte-carlo sampling distribution for 2_1 by following these statements:
1) run setup file "sampling setup"
2) under execute options choose "simulate data"
3) run analysis
4) in sas, submit the file "sampdist"
5) produced a histogram of your sampling distribution
[right now, not working]
Or in unix:
```
     % setup sampdist
     % nice kliqfind yourfile s
     % sas sampdist
```

There should be histogram of the sampling distribution in sampdist.lst. Note that you are essentially running kliqfinder as many times as indicated by the first value in simulate.par (the default is 100), so you may want to run this at night. A network of 80 will take about 15 minutes to run. You may also want to conduct this analysis in a separate directory.

PREDICTED ACCURACY: LOG ODDS OF COMMON SUBGROUP
MEMBERSHIP, + OR - .5734 (FOR A 95% CI)

1.46085

The Log odds applies to the following table:

KNOWN SUBGROUP	OBSERVED SUBGROUP DIFFERENT	OBSERVED SUBGROUP SAME
DIFFERENT	A	B
SAME	C	D

THE LOGODDS TRANSLATES TO AN ODDS RATIO OF

4.30962
WHICH INDICATES THE INCREASE IN THE ODDS
THAT KLIQUEFINDER WILL ASSIGN TWO ACTORS TO
THE SAME SUBGROUP IF THEY ARE TRULY IN THE

```
IN THE SAME SUBGROUP.

              OUTPUT
*********************************************
This file contains the main output.
It is called: stanne.clusters
```

Making the Plots

KliqueFinder has put output into inlist.bicrd and inlist.bcord that indicate the coordinates of the actors. This information, combined with the information in the original list file, is enough to create the plots as described in "Identifying and Mapping Cohesive Subgroups."

The plots are actually constructed by a sas file. In order to produce the plots for the most recent analysis you must type:

click "make graph" (or in unix, % sas socgramz)

The "plot" is now in a posctscript file called "socgramx.eps".
You can download it and look at it using ghostview, or call it into
a word perfect graphics box and print it out on a postsript printer.

Editing socgram.sas:

basically the idea is to look for the variable that controls the feature you want (use ctrl-s to search in emacs)

control over line apperance:

wantline (1=yes, 0=no)

Can also coordinate with boolean expressions based on groupch1 (chooser's group) and groupch2 (chosen's group).

wantarrw (1=yes, 0=no)

ak= controls distance between tip of line and actor ID number

wantwt =1 if you want thickness of line to reporesent weight of exchange

thickd= controls thickness of lines

kline = controls line type (1=solid, 34=dotted)

Control over basic plot:

axis1= controls dimesnions of horizontal, x, axis.

axis2= controls dimensions of vertical, y, axis.

The default is automatic. To change this, look for the commented out versions (with a "*"). Remove the * and edit.

Changeig = **Perhaps the most critical feature.** The smaller the value, the more actors are pulled in towards their subgroup center. If the lines and actors in your first picture are all over the place, decrease changeig. If they are too concentrated within subgroups, increase changeig. Changeig default is 1. Changes distort the dimensionality within and between subgroups. KliqueFinder automatically guesses the best esthetic value of changeig, so check the rescaling factor as printed in *readig*.

Bsize= must look for the *bsize* value associated with the plot command near the bottom of socgram.sas. This controls the sizes of the subgroup boundaries.

Control over plotted locations of subgroups and subgroup labels

useadj = 1 if you want to modify things, 0 otherwise. If you modify, modifications are pulled from the file "inlist.xadjust." See which variables are read in in which order to decide which modifications you would like to make.

wantglab=1 if you want group labels, 0 otherwise.

gdiam controls the size of the circle for a given subgroup.

Control over actor characteristics

Actor characteristics are contained in *cperson2*. You may perform a merge in socgram (look for "merge haveplc." Remove the comment "*" and merge with a file that contains person characteristics that you can then assign to cperson2 in socgram.sas).

Idsize= size of id to be printed in picture.

Wantnum=1 if you want id's printed, 0 otherwise.

To merge with background data:

ID from other sas file

or in socgramz.sas, replace

```
set haveplc;
* merge haveplc  in2.backg (in=havemag);
*by chooser;
```

with

```
*set haveplc;
merge haveplc in2.backg (in=havemag);
by chooser;
```

backg should be the regular sas file that contains actor attributes.

Then set ceperson2= to whatever variable contains the individual characteristics that you want shown in the output.

The basics: All You need to do

In Windows:
Go to pikachu.harvard.edu/wkf and enter kf as the logon. The password is ssdi8733

Follow directions to install.

1) Under basic specs, do browse to choose the directory in which you want to work. Note that "Browse" cannot create a new directory.

2) setup file
    choose "basic setup" and then "Run setup file"
3) Choose a data file (either your own or one of the examples)
4) choose options as described in this manual
5) click "run analysis"
6) look at clusters output, confirm evidence of clusters
to make the picture

7) click "make graph"
8) make modifications based on contents of this manual
9) click run sas (at the moment, this doesn't work perfectly. Sooo .....
Open sas
    1) from the program editor, open the file socgramz.sas in the directory in which you wer working above.
    2) find (ctrl-f) "inlist.axdjust"
    3) modify "inlist.xadjust" to include full path: "c:\...\inlist.xadjust";
    4) find (ctrl-f) "socgramx.eps"
    5) modify "socgramx.eps" to include full path "c:\...\socgramx.eps"
    6) run
10) view file "socgramx.eps" using word perfect, adobe, or ghostwriter.

You may have to edit socgramz.sas (as described in this manual) to modify plots.

In Unix:
A few general comments:

1)All analyses assume you have logged on to edstat2 and changed directories to your directory.
2)A "%" indicates the unix prompt.
3)Remember your emacs cheat sheet commands (which is copied into your directory under the filename *emacs.commands* when you do "setup basic").
4)Files helping you to learn sas are in the directory *learnsas* on edstat2
5)* indicates steps you need to do only once

1. Log on to edstat2.educ.msu.edu under the id *kliqfind*,

2*. create your own directory
% mkdir [directory name]

3. Change to that directory
% cd [directory name]

4*. Set up files
% setup basic
note that you only need to do this once

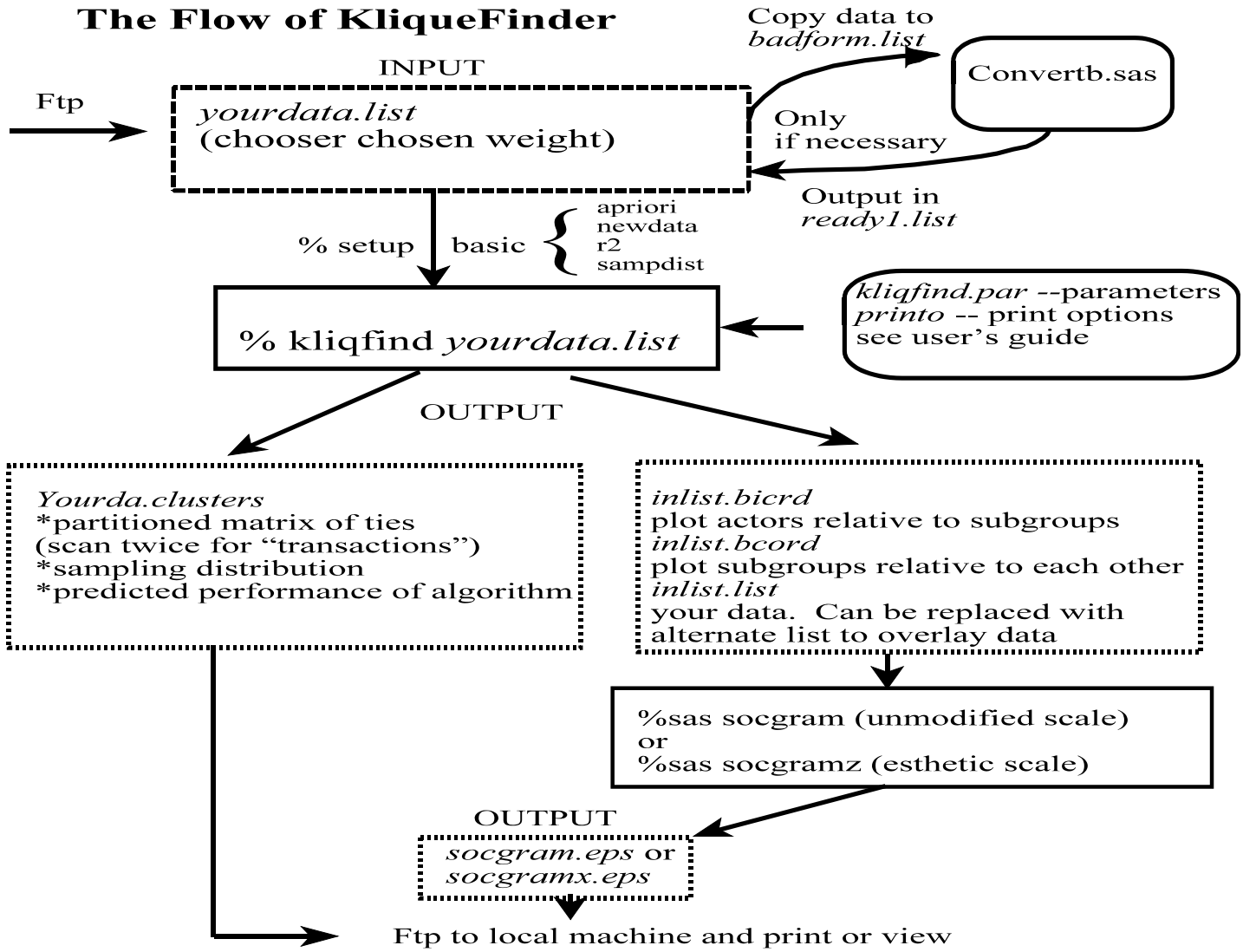
5. Either through ftp or cut and paste, copy your data to a file on edstat2 in your directory
If your data are not in the correct format you will have to run "convertb.sas." – see user's guide.
Sample data are in ffe.list or stanne.xlist

6. Run KliqueFinder to produce clusters and MDS
% nice kliqfind [yourdata]

This will produce lots of output as well as files for making the plots. The main output is in filename.clusters, where "filename" contains the first six characters of your file name.

7. Generate crystalized sociogram
% sas socgramz

The crystalized sociogram is now in a postscript file called socgram.eps (or socgramx.eps)
Ftp this to your pc and print using a postscript printer.



All you need to do:

1) Under basic specs, do browse to choose the directory in which you want to work. Note that "Browse" cannot create a new directory.

2) setup file
   choose "basic setup" and then "Run setup file"
3) Choose a data file (either your own or one of the examples)
4) choose options as described in this manual
5) click "run analysis"
6) look at clusters output, confirm evidence of clusters
to make the picture

7) click "make graph"
8) make modifications based on contents of this manual
9) click run sas (at the moment, this doesn't work perfectly. Sooo .....
Open sas
   1) from the program editor, open the file socgramz.sas in the directory in which you wer working above.
   2) find (ctrl-f) "inlist.axdjust"
   3) modify "inlist.xadjust" to include full path: "c:\...\inlist.xadjust";
   4) find (ctrl-f) "socgramx.eps"
   5) modify "socgramx.eps" to include full path "c:\...\socgramx.eps"
   6) run
10) view file "socgramx.eps" using word perfect, adobe, or ghostwriter.