

# NEGOPY

4.30

## Manual and User's Guide

William D. Richards  
School of Communication  
Simon Fraser University  
Burnaby, B.C. Canada

The production of this manual was supported in part by a grant from the Social Science and Humanities Research Council of Canada.

© 1995 William D. Richards

School of Communication  
Simon Fraser University  
Burnaby, BC V5A 1S6  
Canada

(604) 291-3687  
(604) 291-4024 fax  
richards@sfu.ca

Index: [A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [J](#) [K](#) [L](#) [M](#) [N](#) [O](#) [P](#) [P](#) [Q](#) [R](#) [S](#) [S](#) [T](#) [U](#) [V](#) [W](#) [X](#) [Y](#) [Z](#)

Chapter 1: Introduction to NEGOPY .....	8
Overview .....	8
How NEGOPY categorizes members of a network .....	9
Isolates .....	9
Participants .....	10
A few comments about the category system used here .....	12
Is NEGOPY the right program for you? .....	13
Different versions of NEGOPY .....	15
 Chapter 2: How NEGOPY works .....	 17
Step one: initialization .....	19
Step two: read in the data .....	19
Step three: initial detection of groups .....	22
Step four: formal criteria testing .....	26
Step five: final output .....	29
 Chapter 3: The Data for NEGOPY .....	 30
Background .....	30
Relationships and descriptions of relationships .....	31
Symmetricality .....	33
Transitivity .....	34
distance .....	35
<i>content or function</i> .....	35
strength .....	36
About counting links .....	37
Multi-stranded .....	37
data is obtained by analyzing a videotape .....	38
e-mail usage patterns .....	39
data collected by the diary method .....	39
Measurement .....	42
Assumptions NEGOPY makes about your data .....	43
link strengths are scaled as ratio quantities .....	43
all links represent the same kind of relationship .....	43
requires a single indicator of the strength of each link .....	43
Strength transformations .....	44
Types of strength transformations .....	45
1. Scale reversal .....	45
2. Exponentiation .....	45
3. Combining multiple weights .....	46
4. Computing elapsed time scales .....	47
5. Other ways of combining multiple indicators .....	48

Setting up strength transformation equations . . . . .	48
Examples of valid strength transformation equations . . . . .	49
Examples of invalid equations . . . . .	49
Summary . . . . .	50
<b>Chapter 4: Input and Output for NEGOPY . . . . .</b>	<b>52</b>
Introduction . . . . .	52
Default parameter definition file . . . . .	54
Description . . . . .	54
Altering the DPD file . . . . .	54
The Namelist file . . . . .	55
Description . . . . .	55
Structure of the Namelist . . . . .	55
Link data file . . . . .	57
Description . . . . .	57
Specifying the format of the data . . . . .	57
An example of part of a data file . . . . .	58
Some possible format specifications . . . . .	58
Parameter setting file . . . . .	59
Description . . . . .	59
Header Lines . . . . .	59
Setting a Parameter . . . . .	60
Standard output file . . . . .	61
Description . . . . .	61
Auxiliary output file . . . . .	62
Description . . . . .	62
Contents . . . . .	62
Organization of the file . . . . .	62
Information about each person's links . . . . .	63
Information about the group structure . . . . .	65
Final summary table . . . . .	66
Listing of internal data structure . . . . .	67
Group information . . . . .	67
Node information . . . . .	67
Link information . . . . .	68
Monitoring your run . . . . .	69
Description . . . . .	69
<b>Chapter 5: NEGOPY 4.30 Parameters . . . . .</b>	<b>70</b>
Introduction . . . . .	70
1. Parameters for describing your data . . . . .	71

P01	highestidnumber	71
P02	lowestidnumber	71
P03	Not Used	71
P04	namewidth	71
P05	reciprocation	72
P06	unreciprocateminimum	72
P07	meanstrengthset	72
P08	direction	73
P09	Not Used	73
P10	multiplelinks	73
P11	weakestlink	74
P12	strongestlink	74
2. Parameters that control output		74
P13	132-columnprint	74
P14	nrawprint	74
P15	linkinformation	75
P16	isolateinformation	75
P17	densityhistogram	75
P18	testinformation	75
P19	groupinformation	76
P20	auxfileoutput	76
P21	monitorinformation	76
P22	dmatrixprint	76
P23	onefileoutput	76
3. Parameters controlling group detection and testing		77
24	niterations	77
P25	scanradius	78
P26	sensitivity	78
P27	2linksgivingroup	78
P28	%withingroup	79
P29	minimumsplitsize	79
P30	splitstandarddeviation	80
P31	dropsplitfraction	80
P32	Transweight	80
P33	2-stepweight	81
P34	strength	81
Chapter 6: Output from NEGOPY		82
Introduction		82
A. Initialization and Preprocessing of Data		83
Parameter Settings		84

Listing of Segment of Raw and Processed Data . . . . .	87
Summary of data preprocessing stage . . . . .	88
About counting links... . . . .	88
Processing ID numbers . . . . .	91
Testing Link Strength . . . . .	91
Processing Multi-stranded links . . . . .	92
Processing unconfirmed links . . . . .	94
Processing of confirmed and unconfirmed links . . . . .	94
B. The Links of Individuals . . . . .	95
C. Information about Group Detection . . . . .	101
Structural Differentiation . . . . .	101
The Density Histogram . . . . .	102
D. The Groups as Units . . . . .	104
The Distance Matrix . . . . .	104
Analysis of the Distance Matrix . . . . .	105
E. Individuals in Groups: intensive analysis . . . . .	107
Link Analysis Tables . . . . .	107
Number of outside-group links . . . . .	109
Amount of outside-group interaction . . . . .	110
Strengths of links . . . . .	112
Zone Overlaps . . . . .	112
Group connectiveness . . . . .	113
Links going outside the group . . . . .	114
Final Summary Tables . . . . .	114
The last table . . . . .	115
Chapter 7: On running NEGOPY . . . . .	118
Preparation of Data File . . . . .	119
Creating the Parameter Specification File . . . . .	120
A. Describing your Data . . . . .	120
1. ID Numbers . . . . .	120
2. Directed or Undirected data . . . . .	120
3. Confirmed & Unconfirmed links . . . . .	121
Forcing symmetry . . . . .	122
Strength & Confirmation . . . . .	122
Unconfirmed links . . . . .	122
4. Multi-stranded Links . . . . .	123
5. Number of Links . . . . .	124
6. Strength Indicators . . . . .	124
7. Namelist . . . . .	124
B. Managing the strength of links . . . . .	124

C. Controlling the Output . . . . .	125
D. Controlling Group Detection and Testing . . . . .	126
Using the Density Histogram . . . . .	126
One Big Group . . . . .	130
Very Small Networks . . . . .	132
Using the Distance Matrix . . . . .	133
Running NEGOPY . . . . .	135
Cautions . . . . .	136
Handling unreciprocated / unconfirmed links. . . . .	136
Deleting weak links . . . . .	137
Strength transformations . . . . .	137
Adjusting parameters that control logical testing of groups . . . . .	139
Chapter 8: Performance and Parameter Sensitivity . . . . .	140
Introduction . . . . .	140
Assessment of the program: three views . . . . .	140
Conceptual Issues . . . . .	140
Operational: issues . . . . .	144
Pattern-recognition . . . . .	144
Finding Isolates . . . . .	145
Two stages of analysis . . . . .	145
Parameters . . . . .	146
Parameters that describe your data . . . . .	146
Parameters that control the output . . . . .	147
Parameters that control the definition, detection, and testing of groups . . . . .	148
The first stage of analysis . . . . .	148
The second stage . . . . .	148
Pragmatic issues . . . . .	149
The SK network . . . . .	150
S&K Results . . . . .	151
S&K Summary . . . . .	153
The EQN data . . . . .	153
EQN Summary . . . . .	158
The 249 Data . . . . .	158
249 Results . . . . .	158
249 Summary . . . . .	160
Discussion . . . . .	161
References . . . . .	163
Related readings . . . . .	163
Index . . . . .	172





default settings. If you have unusual analytic needs or unusual data, it may be necessary for you to alter some of these value assignments.

## HOW NEGOPY CATEGORIZES MEMBERS OF A NETWORK

As was explained above, NEGOPY detects groups and assigns individuals in the network to a discrete set of categories.

This set of categories is based on a systems-theoretic approach to organizations. The basis idea here is that systems are comprised of subsystems, which may themselves be comprised of smaller units, etc. At any level of analysis, the distinguishing feature of the members of any unit ("*group*" in NEGOPY terminology) is that the members of that group have more of their interaction with members of their group than with non-members or with members of other groups. Also, if the members of a group are to function together in a coordinated way, there must be some possibility of communication among the members of the group. Besides the members of groups, there are often individuals who do not belong to groups, but who function as intermediaries, to provide connections between groups. Finally, there are individuals who are minimally linked to the rest of the system. The category system used by NEGOPY follows directly from these ideas:

A *Group* is a set of at least three individuals which satisfy the following criteria:

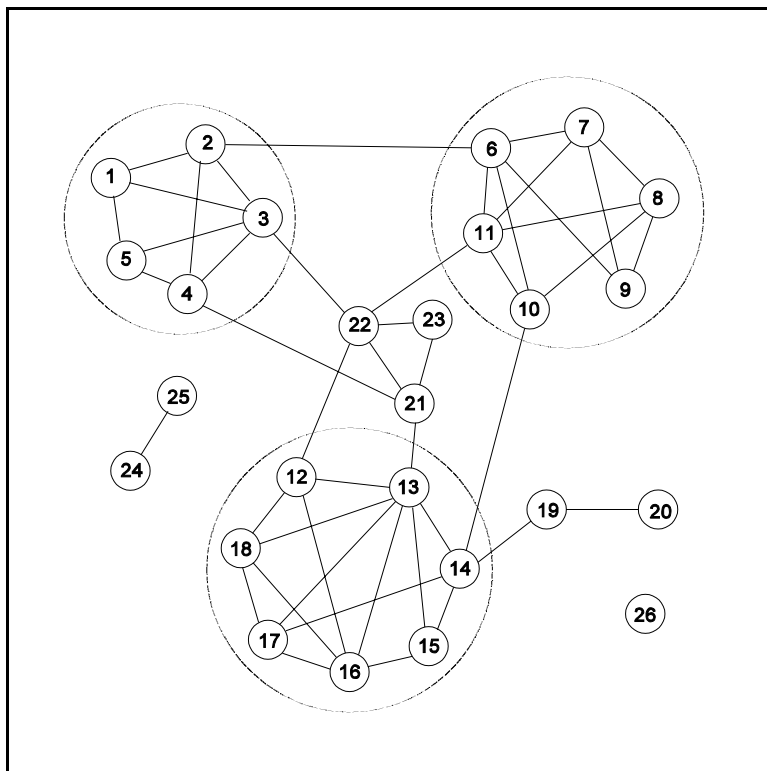
- More of the linkage of each member of the group is with members of the group than with non-members.
- There is some path, lying entirely within the group, from each member to all the other members. (The group is *connected*).
- It is not possible to cause the group to become disconnected by removing ten percent of the members of the group. (This criterion prevents situations in which two or more groups with a few connections between themselves are identified as a single group.)

There are two major categories into which individuals are assigned -- *Isolates* and *Participants*. Each of these major categories is broken down into a few sub-categories:

- **Isolates:** these include all the individuals who are minimally connected to others in the network:



3. **Liaison Type 2 or Multi-step Liaisons** -- these are individuals who do not have most of their interaction with members of groups. They provide indirect or 'multi-step' connections between groups by connecting Liaisons, who have direct connections with members of groups.



**Fig. 1** NEGOPY roles

Group members:	--	#1 through #18
Liaisons: (direct)	--	#21 & #22
(indirect)	--	#23
Isolates: (type 1)	--	#26
(type 2)	--	#20
(dyad)	--	#24 & #25
(tree node)	--	#19



eight links, three of which are with the members of a group, can be a member of that group if those three links account for more than 50% of that individual's total interaction with participants.

## IS NEGOPY THE RIGHT PROGRAM FOR YOU?

NEGOPY **may** be appropriate if your data satisfies these criteria:

- You have descriptions of pairwise links between the members of a social unit of some kind.
- You have data from *all* (or almost all -- at least 90%) of the members of the unit (i.e., you do not have data from a sample of members).
- You have data for all of the links each individual has with others in the network. (i.e., your data collection instrument was not designed to accept only the 3 strongest links each person has, etc.)
- For each link you have an identification number for both of the individuals involved in the link. You may also have some indicators of the strength, content, mode, etc. of the link.
- The social unit for which you have data is comprised of between 25 and 4000 individuals. (The maximum size of network you can analyze at your installation may be lower than 4000, depending on the computer you are using.)
- No individuals (or only a *very small* number of individuals) have links to all or most of the other individuals in the network. If there are individuals with links to everyone, it is not likely that NEGOPY will find any groups, because there will be no groups, according to the definitions the program uses.
- If you do have strength weights for links, they should be scaled in such a way that at least a rough approximation to ratio-level estimates of link strength can be obtained. The critical issues here are first, that a link with a strength of zero is considered to be the same as no link at all; and second, that a link with a strength of  $2X$  is considered by NEGOPY to be twice as strong as one with a strength of  $X$ . If your strength weights do not meet these criteria and cannot be made to meet them by means of strength transformations which NEGOPY can perform, you may still wish to use the program if you can justify treating



of the hierarchical clustering methods.

- Finally, you should not use NEGOPY if you do not like the way it defines groups, liaisons, etc., although even in these situations, you may find the program to be a useful pre-analysis tool for learning more about your data.

## DIFFERENT VERSIONS OF NEGOPY

There are several different versions of NEGOPY in use today. They can be divided into four major categories:

1. NEGOPY-1 -- the CDC version that was released between 1971 and 1978. *This version is no longer supported.*
2. All mainframe IBM versions that do not have the letters "EQN" in their name are referred to as NEGOPY-2. The IBM FORTRAN G versions (with dynamic run-time memory allocation) were released from 1979 to 1986; the FORTRAN 77 versions (for IBM, CDC, VAX, etc.) were first released in late 1985. NEGOPY-2 has many improvements over the older CDC version, including better control of the output which is formatted to be more easily utilized. All of these versions have 45 parameters. *This version is no longer supported.*
3. The first EQN versions were written in FORTRAN G for IBM mainframes. They were more advanced than both NEGOPY-1 and NEGOPY-2, offering better performance and greater ease of use. The EQN versions eliminate all of the parameters that were required for the Strength Transformation Equation. (The user specifies the equation by writing it out in ordinary arithmetical notation.) Although the EQN versions have only 33 parameters, they provides additional options for pre-processing the data before analysis and for the handling of directed links. The EQN versions are referred to as NEGOPY-3 or NEGOPY- EQN. *This version is no longer supported.*
4. Finally, the focus of this manual: a FORTRAN 77 version of NEGOPY for the IBM PC, first made available in early 1987 and most recently revised in 1993. This version has numerous improvements over mainframe NEGOPY-3, including more control over output options, more automatic





## Chapter 2: How NEGOPY works

*While it is not necessary for you to memorize all the details of this section, a basic understanding of the procedures used by NEGOPY will help you use the program more effectively and avoid making errors that may invalidate your results.*

For data, NEGOPY wants a list of all the links between the members of your network. For purposes of description, each individual in the network is assigned an ID number. A link is described by providing the ID numbers of the two individuals connected by the link and some additional (optional) information about the link. Throughout the analysis, individuals in the network are referred to by means of their ID numbers. Generally, these ID numbers should range from **1** to **N**, where **N** is the number of individuals in the network.

NEGOPY uses a two-stage process to identify groups. In the first stage, a rather messy pattern-recognition procedure is applied to the results of an iterative process that rearranges the data in such a way that any groups that exist become "visible". In the second stage, a number of logical tests are applied to the tentative description provided by the first stage, to make sure that the description meets the formal criteria specified in the definitions of role categories. It is important to note that the results of the first stage are only tentative, and that a good deal of "cleaning up" must be done in the second stage before results can be reliably interpreted.

The program can be divided into five more-or-less distinct stages:

1. Initialization -- Read control parameters and allocate storage for data.
2. Read in Data -- Data are read; strength transformations (if any) are performed; and the initial data structure is set up.
3. Initial Group Detection -- The iterative averaging process is completed; the data structure is elaborated; and the first tentative description of groups is formed.
4. Formal Criteria Testing -- The tentative description is refined through the application of logical and mathematical tests; group distance matrices are constructed and analyzed; and the data structure is filled in.
5. Final Output -- Final tables and charts describing the network are produced and printed; summary tables are printed.



*stranded* links may be combined by adding their strengths together, or all but one may be dropped from the rest of the analysis.

Following this sorting procedure, a "reciprocation" analysis is performed. This means that for any link between a pair of individuals, say from **A** to **B**, a search is made for a corresponding link from **B** to **A**. A link for which both "halves" are present is called a "reciprocated" or "confirmed" or "two-way" link. The first two of these three terms may have precise technical meaning (described below), which overlap somewhat with "two-way".

A "two-way" link is simply one that goes both directions. In other words, there is a link from **A** to **B** as well as one from **B** to **A**. A "one-way" link goes in only one direction.

Strictly speaking, confirmation and reciprocation are quite different things. If I say I give you money and you say I give you money, my description of the relationship is *confirmed* by your description. Our descriptions are in agreement with one another. If, on the other hand, I say I give you money and you say you give me money, we have a description of a *reciprocated* relationship: What I do to you, you also do to me.

Confirmation and reciprocation will be the same when the relationship under consideration is *symmetrical* or *un-directed*.

Communication relationships are generally assumed to be undirected relationships. If I communicate with you, you also communicate with me. But "information giving" and "information receiving" are directed relations. With undirected relationships, reciprocation and confirmation amount to the same thing.

If the relationship you are studying is directed, you will need a separate measurement instrument for both halves of the action if you want to confirm your links. If you want to study information or advice giving and receiving networks, you will need to collect separate data for "to whom do you give information" and "from whom do you receive information." It is likely that a comparison of the two networks described by these two sets of data will reveal profound differences. (NEGOPY does not have the capability to handle this kind of "full duplex" data.)

The program determines which links are confirmed/reciprocated, and, depending on how you have set **P05 (reciprocate)**, and **P06 (unrecipmin)**, it either deletes unconfirmed/unreciprocated links, or it adds markers so the program (and you) can



### **STEP THREE: INITIAL DETECTION OF GROUPS**

NEGOPY uses a two-step process for the detection of groups. The first part involves a graphic pattern-recognition process to obtain a tentative approximation of the group structures. Because this process is not exact, there is a second stage, in which a number of logical criteria are applied. The first (initial or tentative) stage is explained in this section.

The **tentative group-detection process** is also comprised of two stages. The first involves the application of an iterative process which makes groups visible. The second involves the application of another process which actually identifies the groups.

The **iterative process** is based on the idea that the members of a group talk more with one another than they do with members of other groups. The goal is to bring members of groups close to one another, in tight clusters, so that the clusters will be visible. The procedure works as follows:

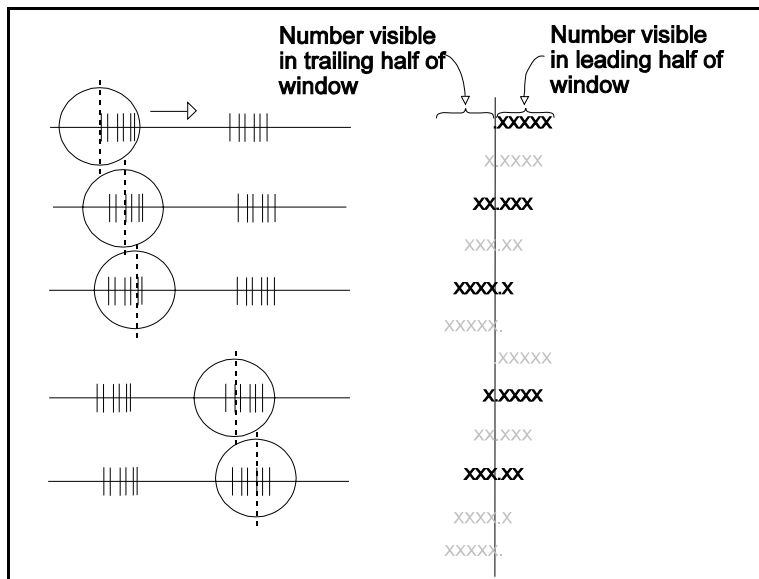
- Individuals are spread out along a line or continuum  $N$  units long (where  $N$  is the number of members in the network). They are located on the basis of their ID number, so that the person with the lowest ID number will be at one end and the person with the highest number will be at the other. From now on in the description of the procedure, individuals will be referred to in terms of their "location", which is a point on the continuum. For each individual a mean is calculated. This is the mean of the location of this individual and of the locations of all the individuals this one has links with.
- Each individual is moved to a new location, determined by that individual's mean.
- A new mean is calculated for each individual, based on the new locations of that individual and those with whom this individual has links.
- Individuals are moved to new locations, etc...

Because the strength of links is taken into consideration in the definitions of groups, the strengths are used as weighting factors in the calculation of means.



The second phase of the tentative group detection process involves the application of a **pattern-recognition** technique which results in a determination of how many groups there are and which individuals belong to which groups. This process works as follows:

- A "window" is constructed and scanned down the length of the continuum, stopping over each individual on the continuum.
- At each stop, the number of individuals visible through the leading and trailing halves of the window is counted. These numbers are displayed graphically in the "Density Histogram" in the printout produced by the program.



**Fig. 2** The scanning window in operation. The figure shows the window in the 1st, 3rd, 5th, 8th, and 10th positions.





seems to be drawing a large number of group boundaries and finding many very small groups, or if it does not draw enough boundaries, so it finds only a very small number of quite large groups, this parameter might be adjusted.

*Many networks do not break apart into "nice" group structures, and it is quite likely that the effects of adjustments of these parameter will either be effectively cancelled in later stages of processing the data, or that they will prevent the program from identifying any groups that might exist. The default values have been chosen after evaluating extensive test results, and have been shown to be appropriate in most situations.*

## STEP FOUR: FORMAL CRITERIA TESTING

It is possible for there to be a number of kinds of problems with the results of the tentative group identification process described above. Some members of groups might not be put in groups and others might be put in the wrong groups. Some individuals who are not members of groups might be put in groups. What is identified as one group might actually be two or more groups. The procedures described in this section are designed to identify and correct these problems. The procedures include three tests that are applied to individuals: two for individuals not placed in groups and one for individuals that are placed in groups. There is also a set of tests that applies to groups as units.

**Note:** It is the *amount of interaction, rather than the number of links* that determines the assignment of individuals to network roles. Thus a single link with a strength of 4 will be considered to represent as much interaction as two links, each with strengths of 2.

The first test is to examine each individual not assigned to a group and to determine whether or not most of the individual's interaction is with the members of any existing group. If it is, the individual is added to the group.

The second test is to take all the individuals not placed in groups and see if a group can be formed from them. This is done by placing all non-group members in a



average, to the rest of the members than will the others. In other words, if there are more than one group stuck together, there will be a greater amount of variability in the distribution of mean distances than if there is only one group. The standard deviation of the column means is thus used as an indicator of the likelihood that there is more than one group present. If the standard deviation is greater than a set threshold value, an attempt will be made to split the group into its sub-groups.

The splitting procedure works as follows:

- The list of column means is rank-ordered from lowest to highest.
- Ten percent of the group's members -- the ones with the lowest means -- are temporarily removed from the group.
- A new distance matrix is calculated.
- If the group is still connected, the removed members are put back, and the group is considered to have passed the test. If, however, the group is no longer connected, a split is performed. The "fragments" of the group are examined. If they are too small, they, together with the temporarily removed members, are put back into the original group.
- If the fragments are large enough, the members temporarily removed are reclassified as liaisons, and two (or more) new groups are formed.

A number of aspects of these procedures can be controlled by means of the parameters:

- Many tests require that "most" of an individual's interaction be with other members of the same group. **P28 (%within)**, specifies exactly the minimum amount of interaction this "most" must be. The default is set to **50.01%** -- just over one half. You may raise the value of this parameter if you have a conceptual or theoretical justification for doing so.

*Note that sufficiently high values may prevent any groups from being detected.* The value of this parameter *should not* be set to values below the default of **50.01**. Doing so will result in ambiguous situations in which it is possible for an individual to belong to more than one group. NEGOPY does not have the ability to handle multiple membership. The results will be *unreliable and probably*



## **STEP FIVE: FINAL OUTPUT**

After the groups and individuals have all been tested, NEGOPY prints full descriptive summaries of the group structures and of the linkage of group members. A summary table is produced for each group member, showing breakdowns for within-group links, between-group links, and links with liaisons. More detailed summaries are produced for each group. There are also lists of all the *bridge* (direct contacts with members of other groups) and liaison contacts of each group. Following the group listings, there are summaries of the links of liaisons and isolates. There is also a final summary list that shows all the members of the network ordered according to ascending ID number.



description throughout the analysis (i.e., that you don't do inappropriate things to your data). In this respect NEGOPY is really no different from any other method for analyzing data.

A second point to come from the distinction between the system and the network has to do with the way relationships between individuals in the system are represented as links in the network. Relationships are complex and multidimensional. They are also highly dynamic, as they change more-or-less constantly over time. While many network analyses describe the relationship under investigation as something like "talks-with", as in "who talks with whom", the simplicity here is only apparent. Bernard, Killworth, and Sailor (BKS) have conducted a program of research over a period of several years that indirectly points out the complexity of this situation. Their research conclusion is that people are not able to accurately say who they talk with, and that there is little relation between what people say they do (in terms of communication) and what they actually do. A careful reading of the published reports of this team leads to the conclusion that communication relationships are indeed much more complex than they seem, and that it is very difficult to obtain accurate measurements of these relationships. For this reason it is important to have a thorough understanding of what kind of relationship it is that you are studying. This understanding will help you to make intelligent decisions about how to collect, represent, and analyze the data. The following discussion will give an introduction to some of the complexities one runs into in attempting to obtain valid network data.

## **RELATIONSHIPS AND DESCRIPTIONS OF RELATIONSHIPS**

As was pointed out above, NEGOPY begins with a description of the system, in terms of the pairwise relationships between the members of the system. Because the data are only *descriptions* of relationships, the data must be quite abstract and, to a degree, arbitrary, in comparison to the system itself, which exists on a much more concrete level of reality. The distinction being made here refers to the difference between the reality of the situation and how that situation is described by the data -- that is, what aspects of the system are included in the data, how the data are coded and scaled, and the limitations imposed on the fidelity of the description provided by the data. The peculiarities of descriptions turn out to be crucial to the process of network analysis, for the logic of descriptions is tangled with the logic of relationships in complex ways.

Consider, for example, the social relation that exists between a pair of friends who work in





the kinds of relationships that may exist between and among people. On the other hand, there are so many restrictions on the kinds of links that may be used in network analysis that it is much easier to say what is possible, rather than what is not. One key restriction is that only *pairwise* relationships -- those involving two people at a time -- are considered. If there are three people involved, there must be at least three relationships for purposes of analysis: one between Floyd and Magoo, one between Floyd and Ellen, and one between Ellen and Magoo, even though this pairwise description cannot possibly capture the true nature of the triad. At the abstract level of description currently used with linkage-based methods of structural network analysis, the only qualities of relationships between people that survive are *symmetricity*, *transitivity*, and *strength*. Sometimes a form of "content" or "functional" categorization is also permitted.

**Symmetricity** is one of the most prominent abstract characteristics of relationships that must be considered in almost all approaches to network analysis. Symmetricity is related to direction or order. In a *symmetric* relation, if **B** is related to **A**, **A** must also be related to **B**. A symmetric relation has no direction -- it is like a two-way street. In an *asymmetric* relation, the fact that **B** is related to **A** does not imply that **A** is related to **B**. In a *strictly asymmetric* relation, if **A** is related to **B**, **B** cannot be related to **A**. A one-way street is an asymmetric connection between points along its length -- which can cause difficulty if you are at point **A** and are in a hurry to get to point **B**, which is in the wrong direction.

- **Symmetry** should be distinguished from *confirmation* and *reciprocation*, two closely related ideas. Symmetry is a property both relationships and data may have, while confirmation and reciprocation are aspects of measurement. If I report a relationship with you (e.g. I say I give you information) and you support my report of the relationship (e.g. you agree that I give you information), the report is said to be "*confirmed*". If you do not report the relationship, my report is "*unconfirmed*". If I say I have a particular kind of relation with you (e.g. I trust you), and you say you have the same kind of relation with me (e.g. you trust me), then the me-to-you link is "*reciprocated*" -- you do the same thing to me that I do to you. When the relationship under investigation is symmetric, reciprocation and confirmation will amount to the same thing. When the relation is non-symmetric, however, reciprocation and confirmation are quite different. Take, for example, the relation "trusts". If I say "I trust you," and you say "Bill Richards trusts me," my report is confirmed. It will not be reciprocated, however, unless you say you trust me.
- Symmetry also applies to data. Network data can be arranged in a matrix, where there is a row and a column for each member of the network. Ordinarily the rows are for



to her in my life? Questions like these cannot be answered unless some understanding of what the individual does with information is built into the conceptual model of the relationship. While this type of transitivity may have some very interesting applications, it is rarely the explicitly focus of network research.

Network researchers often refer to the "*distance*" from one node to another. There are at least two different senses of the term "distance":

- many analytic techniques (e.g. factor analysis, and some other forms of multidimensional scaling) use the term to mean the opposite of "*similarity*", where two nodes that have similar patterns of connection to other nodes in the network will be said to be "similar" or "close" to one another (in some usually unspecified multidimensional space);
- there is also the distance from one individual to another, defined as the *number of steps* it takes to get from node X to node Y. Here a pair of people connected directly are one step apart. If the first person has a contact with a second who has a contact with the third, the first and third will be two steps apart.

The first type of distance is not used in NEGOPY. The second is, and it is of interest in this discussion because it raises issues related to the concept of transitivity. It makes little sense to speak of distance in terms of how many links must be passed through, unless the relationship which defines the links is transitive in the mathematical sense. If the relationship is intransitive, the fact that I am linked to you and you are linked to Nancy says nothing whatsoever about the nature of the relation between Nancy and me.

The *content* or *function* of the relation creates some of the messiest problems in network analysis. The interaction between content and transitivity was briefly alluded to above. If **A** is connected to **B**, you would want the same kind of thing happening in the **A-to-B** step as in the **B-to-C** step before you would be willing to say that there is an implied or indirect relation between **A** and **C**.

Similarly, it makes sense to say that the same kind of thing should be happening in all the links in a network for it to be a single network. Indeed, when networks involving different kinds of relationships are mixed together, any structure that might be evident in any of the separate networks is likely to be obscured.

- A common content/function category system for communication networks is Berlo's Production-Innovation-Maintenance trichotomy, which is repeated in Farace et al's Work-Innovation-Social. In a study of communication patterns of foster parents in



“proximity” and “strength” interchangeably. NEGOPY does not refer to the number of shared contacts as “strength” or “weakness”. It calls this either "the number of shared contacts" or the "overlap of 1st-order zones".

### ABOUT COUNTING LINKS...

There is a good deal of confusion about how to count links. Do you count the number of pairs of people who are connected, or do you count the number of reports of connections? Is a reciprocated link counted as one link or two? If the data contains three reports of a link from **A** to **B** (one for production-related contact, one for social contact, and one for contact about questions and changes), how many links are there between **A** and **B**? These issues can be resolved by distinguishing between *links* and *link components*.

A *link component* is the information you get when one individual reports a link with another individual. A *link* is a connection between a pair of individuals, *regardless of the number of individuals reporting the connection*. Therefore, if one individual reports a connection with a second, there will be one link component and one link. If the second individual also reports a connection with the first, there will be *two* link components, but still only *one* link.

If there are three reports of a connection from **A** to **B**, describing three different conversations, each appearing on its own line in the data file, there will be three link components, but still only one link, since only one pair of individuals is connected. In this case the link from **A** to **B** is called *multi-stranded*.

**Multi-stranded** is close in meaning to, but not exactly the same as, "multiplex". A *multiplex* tie between a pair of persons means that the persons relate to one another in different ways. They may have a formal relation to one another at work; they may be close friends; and they may be members of the same hockey team. To say a link is *multi-stranded* is to say that the same relationship between the pair of individuals is described more than once in the data. This may be due to error (the same information was entered more than once more than once in the file, like like this this). Or it may be due to the fact that the measurement and coding procedures made it appropriate or necessary, in which case these multiple links would not be errors, and should not be treated as errors.

With certain measurement approaches and coding strategies you may expect to have multi-









column-->	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	...
	1	0	1		1	0	2		0		2	1				0								7		
	1	0	1		1	0	3		1	5		0				0								0		
	1	0	1		1	0	5		8		1	2			5									0		
	1	0	2		1	0	1		0		2	2		1	1									0		
	1	0	2		1	0	4		0		1	5			9									0		
					ID1		ID2			a				b					c				d			

**Fig. 4** Part of a link data file

- Figure 4 shows a different method of coding the kind of data shown in Figure 3. Rather than putting each interaction on a separate line and indicating communication medium by a second variable, this example uses a single line of data with several columns for each pair of individuals. The four variables after the ID numbers describe the amount of interaction that used the four communication media. The first line, for example, shows that #101 had no face-to-face interaction, 21 minutes of telephone interaction, no e-mail, and 7 minutes of fax (whatever that may mean). With this method of representing the information in the data file, it becomes quite easy to choose a single medium or to create a combination of more than one media. To obtain the total amount of face-to-face and telephone interaction, for example, the following equation would be used:

$$\text{time} = a + b$$

With this method of coding the data, multi-stranded links would probably not be expected. What would it mean, for example, if there was a second line of data describing interactions from #101 to #102? If all the interactions between each pair of individuals are supposed to be included in one line of data, then multi-stranded links would be interpreted as data entry errors. In this case, they would probably not be added together, and the second #101-to-#102 line would be ignored.

The way of counting links outlined here will eliminate confusion and will be consistent with aspects of network analysis (e.g., calculations of density or integrativeness) that are interested only in whether or not there is a connection between a given pair of



amount of information is greatly reduced and altered — if not in content, at least in form.

People do not walk around with descriptions of their social relationships clearly formulated, ready to be listed out on request, the way a computer can list out the contents of its files. The sociometric question is a complex stimulus which demands a significant amount of information processing as individuals attempt to construct from their memories, present feelings, moods, etc. — which are not stored as digital streams of linguistically-encoded symbolic information — a highly structured, abstract, discrete representation of their social realities.

The sociometric question can be seen as stimulating the encoding of memories, feelings, beliefs, etc., into a discrete symbolic format -- either as a list of social contacts, in response to an open-ended question; or as a series of responses to a multiple-choice type roster. This encoding activity is complex, first in that tremendous amounts of emotional, iconic and linguistic information must be mapped onto a coding system many orders of magnitude simpler; and second, in that every individual has a unique way of going about the process. Simply asking respondents to indicate who their contacts are ignores the complexity in this measurement process.

I presented an argument for the need of more sophisticated approaches to measurement in Richards (1985), and tested this approach in 1986. The study began with a series of extensive focus-group sessions with three different samples of potential respondents. These sessions were recorded and transcripts were produced and content analyzed. The result of this analysis was a typology of the types of relationships the individuals have with one another and which was used as the basis for the data collection instrument.

The problem with this approach is that it produces more complex data than is commonly used in network research. Different approaches to analysis become appropriate when each link is described by two dozen variables, in addition to the standard frequency and importance.

*You may save yourself a lot of trouble if you carefully read the preceding paragraphs and consider the relationship you are attempting to study in your research before you design a data-collection instrument.*



and insert it anywhere after the labels in your parameter-setting file. The strength indicator variables are represented in the equation as **a**, **b**, **c**, etc.

*If you don't want transformations ...*

NEGOPY is set up to be as easy as possible to use. If you do not want to make any transformations, you do not have to set any parameters or provide a complicated equation. If you do not want any transformations, use the following strength equation for data with one strength link:

$$\text{strength} = a$$

*If you do want transformations ...*

Just write the equation you want to use and place it in your parameter-setting file. An example is shown in Figure 5 below.

```

This is the 1st title label.
This is the 2nd title label.
highestid = 200
nweights = 3
strength = a * (b + c)
weakest = 10
strongest = 200

```

**Fig. 5** Strength equation in Parameter Setting file

## TYPES OF STRENGTH TRANSFORMATIONS

The goal of all strength transformations is to obtain a single ratio level indicator of the strength of each link. This indicator should have a value of zero if the link has no strength and higher values for links of higher strength. The following paragraphs will describe a number of typical transformations, and how they could be done with NEGOPY.



Frequency	code	times per month	code squared
about once a month	1	1	1
about once a week	2	3-6	4
two to three times a week	3	8-12	9
almost every day	4	14-18	16
once a day or more	5	20-30	25

**Fig. 7** Approximating ratio-level scaling

Squaring the original values can thus be a useful way of adjusting the scaling on an improperly-coded frequency scale. If the variable that contains the coded value is **a**, the transformation equation would be:

$$\text{strength} = a * a$$

### 3. Combining Multiple Weights

It is common practice to collect multiple strength indicators for links. Probably the most common combination is Frequency and Importance. The idea is that both frequency *and* importance must be taken into consideration, since very important interactions that happen once a month are often "stronger" than unimportant ones that happen more frequently (but see the "Strength of Weak Ties" literature, especially Granovetter, 1973). The question becomes how to combine the two strength indicators.

Assume you have two indicators, "frequency" and "importance", and that each is coded from 1 to 5, where 1 means low frequency or importance and 5 means high frequency or importance.

A matrix like the one in Figure 4 could be constructed. All you have to do now is replace the question marks with numbers that indicate the relative strengths of the links. One approach that is commonly used is to multiply the two codes together, to produce the values shown in Figure 5. The transformation equation used to get these results would be:

$$\text{strength} = a * b$$

This combines the two scales well enough, but it doesn't incorporate the previous correction made on the frequency coding. This correction can be made to the combined scales by squaring the frequency values before multiplying by importance. Figure 6 shows the results.





## 5. Other Ways of Combining Multiple Indicators

In other situations you may want to add several indicators, applying a suitable weighting factor to each one. This might be the case if you have indicators for the amount of time spent in face-to-face contact and telephone contact for each link. You might say that face-to-face is stronger than telephone because of the availability of visual cues. If you gave the two modes relative weightings of 2.0 (face-to-face, coded as "a") and 1.0 (telephone, coded as "b"), you might combine them with an equation that looks something like this:

$$\text{strength} = (2.0 * a) + (1.0 * b)$$

More complex transformations, combining scale reversal, differential weighting, exponentiation, etc., can be performed if necessary. The program gives you a great deal of flexibility in this area.

## SETTING UP STRENGTH TRANSFORMATION EQUATIONS

Both the EQN versions and the older versions of NEGOPY have provisions for working with strength weights. With the EQN version, you specify the strength transformation equation in a straightforward written form. With the older versions, you customize a built-in strength transformation equation by setting parameters that control the value of constants and coefficients. Since the older versions are no longer supported, only the EQN version is described here.

With NEGOPY-EQN you simply include a line *in your parameter setting file* that specifies the equation. This line might look something like this:

$$\text{strength} = a + (2*b) + (c-(10*d))$$

There are a few restrictions on the equation:

- It must go *in the parameter setting file, after the two label lines.*
- There may be as few as none or as many as ten strength indicators.
- Indicators must be represented in the equation as *lowercase* letters. The first indicator that appears in the data should be referred to in the equation as "a". The second is "b", and so on.
- The equation begins with "strength ="



$\text{strength} = ((5-a) * b$	Parentheses must be balanced.
$\text{strength} = b / (2-a)$	This may look okay, but if the value of "a" is ever "2", the denominator will become zero. Division by zero is not allowed because the result is infinitely large.
$\text{strength} = (a^{**2}) + (b^{**2})$	If the intent is to say that strength is the sum of a squared and b squared, the symbol "^" should have been used instead of "**", which doesn't mean anything to NEGOPY
$\text{strength} = (a - b) ^ 0.5$	This tells the program to take the square root of the quantity "a - b". If b is ever larger than a, the quantity will be less than zero, making the root imaginary.
$\text{strength} = \sin(a)$	NEGOPY failed trigonometry in high school and didn't repeat the course.

## SUMMARY

This chapter has described several important aspects of the data that can be analyzed by NEGOPY. It began with a discussion of the difference between the system and the network and moved on to a description of the abstract characteristics of relationships that are considered in network analysis. The way links are described was explored, and a discussion of different methods of scaling was undertaken. Finally, the section ended with an explanation of the program's ability to transform coded data so it would meet the assumptions and requirements of the program.

A word of caution is in order: Although NEGOPY makes it easy for you to get more out of your data, it also makes it easy for you to obliterate good information or introduce spurious information. Relational data is complex. It is important that you make sure you understand your data and how it is scaled (coded), and what assumptions are implied by the method of scaling you used. A failure to do this opens the door to invalid, unreliable results.



- **The "link data" file** contains the data to be analyzed by the program. This file describes the links between the individuals in your network. *You provide this file.* The program asks you for the name of your link data file immediately after it asks for the parameter setting file. An example on the distribution disk is DATA.249.
- **The "standard output" file.** This is where the standard print output goes. *The program writes on this file.* Normally there will be only one "standard output" file. But you have the option of having the program break the output file into three smaller files. This option is explained at the beginning of the section of the manual that describes the program's output (Chapter 6). If your network has more than 100 members or 1000 links, you should use either a hard disk or a high-density diskette for the standard output file. The program asks you for the name of your output file at the beginning of a run.
- **The monitor** is an *optional* output device used to describe the operation of the program during execution. It is automatically connected to the program. You control how much information is displayed on the screen by the way you set Parameter 21 (**monitorinfo**).
- **The "auxiliary output" file** -- an *optional* file on which data are written by the program for further analysis. *The program writes on this file.* If you tell the program you want to use an auxiliary output file (which you do by setting Parameter 20, (**auxfileout**), to a value higher than 1), it will ask you for the name of the "Auxiliary Output File" when it reads your parameter settings. You should put this file on a hard disk or a high-density diskette.



## THE NAMELIST FILE

### Description

A *namelist* is a file that contains some kind of alphanumeric (i.e., letters and/or numbers) information that is associated with each individual in your network. This information is printed out whenever the individual is mentioned in the results of analysis. The file is called a "namelist" because the most common use for the file is to associate the names of individuals with the individuals' identification numbers in a network run.

A namelist is a file containing one alphanumeric string for each node. This string will be printed alongside the ID number of the node whenever the node is mentioned in the printout. This string may be up to 20 characters in length. It may contain the name of the person corresponding to the node, or it may contain any other identifying information.

The program will ask you for the name of the "Namelist file" if you set Parameter 4, (**namelist**), to a value higher than **0**. Just type in the name of the file when it asks you this question.

### Structure of the namelist

The structure of the namelist file is shown in Figure 1 on the next page.

- The namelist begins with a fortran format specification.
- There are two fields here -- the ID number, and the name. A good format to use is:

(I3,X,A)

That is, each line of the file will contain a three-digit ID number, followed by one blank space, and then the "name" information. This is the format used for the file shown in Figure 1 on the next page.

- It doesn't matter how many columns of information you have for each name -- you describe the name simply as "A". That is, you use "A" for the names, regardless of how many characters there are in the names. (The value you give for the name-width parameter, P04, tells the program how many columns to read for each name.)





## LINK DATA FILE

### Description

The program will ask you to supply the name of the file that contains the data you wish to analyze. Type the name of this file in response to the prompt.

### In general...

- *Each line* must begin with the *ID number* of the respondent.
- All the currently supported versions of NEGOPY allow only *one* link per line of data.
- The order of links or lines does not make any difference to NEGOPY, but your life may be a bit easier if they are arranged into some kind of convenient order.
- If there are more than one link between a pair of people, only one will be used in the analysis, unless you set the Multiple Links parameter (**P10, multilinks**) to a value of one, which will cause multiple links to be combined by adding the strengths together.

Each line of data must begin with the ID number of the respondent, and must also contain some information that describes a link:

#### **ID#1 ID#2 (weights)**

- **ID#1** is the ID number of the respondent. The respondent is the person reporting the link. This ID number must come *before* the part that describes the link.
- **ID#2** is the ID number of the person the link goes to.
- **(weights)** are strength weights. There may be no weights at all, or up to 10 weights per link.
- all links should have the same number of weights

### Specifying the format of the data

NEGOPY wants its data to be in "fixed" format. That is, values must always appear in the same columns. If the respondent number of the first line of data is in columns 2 to 4, as in the example in Figure 3, the respondent numbers on all lines of data must be in columns 2 to 4. The same is true of the other ID number and any weights you may have. NEGOPY uses FORTRAN format specifications to describe the layout of the data.

### An example of part of a data file

The first few lines of a sample data file are shown in Figure 3. There are two strength



## PARAMETER SETTING FILE

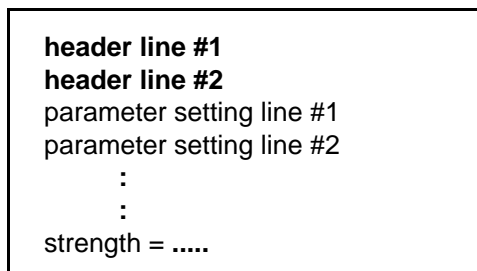
### Description

The operation of the program is governed by a number of parameters. All the parameters have "default" values which are automatically assigned when the program begins to execute. Usually you will want to change the values given to several of the parameters in order to describe the characteristics of your data and to select some of the options available to you. Setting the values of parameters is done with this file. With this file, you indicate which parameters must be set to values other than the default values, and what those new values should be.

The program will ask you for the name of your "Parameter Setting file" at the beginning of each run. Type the file's name and press enter. Include the full path to the file if it is in a directory other than the one you were in when you started the program. (The file name, together with the path, must not exceed 24 characters.)

### Header Lines

The first two lines of the parameter setting file are used for information that describes the run. You might include the date of the run, who set it up, or any other information that will help you know what this run is when you look at the printout sometime in the future.



```
header line #1
header line #2
parameter setting line #1
parameter setting line #2
:
:
strength = .....
```

Fig. 4 Structure of parameter setting file

The header lines are required. If you do not put two header lines before the parameter setting lines, the first two parameter setting lines will be treated as header lines, rather than as parameter setting lines. The contents of the header lines are printed from time to time throughout the output.



## STANDARD OUTPUT FILE

### Description

The program will ask you for the name of the "Standard Output file." Type the name of the file you want the program to use in response to this prompt. The program will tell you if the file whose name you type already exists, and if it does, will ask if it is okay to lose the information in the present file. If you answer "No", the program will prompt you for a different file name.

Because the program produces so much information about your network, you have the option of having the program split the output file into three separate files. If you leave **P23 (onefile)** set to its default value of **1**, you will get only one file. If you set it to **3**, the program will split the output into three files. This procedure is automatic; the program will create the files and choose names for them.

If you give the name "out1.xxx" for the standard output file, the second and third files will be "out2.xxx" and "out3.xxx". To make the names for the second and third output files, the program takes the last character before the "." and adds one. Thus, "outa.yyy" will produce "outb.yyy" and "outc.yyy". The contents of these files are described in Chapter 7.

The output files can be quite long. It is a good idea to have the program put them on the hard disk.

The output and its interpretation are described in Chapter 7, "Output."



- b. A one line title "Parameter settings for this run".
- c. A listing of all the parameters and their values. The format is "PXX=YYYYY" where XX is the parameter number and YYYYY is the value.

```

a. header: Demonstration/verification run for PC-EQN 4.30
           Created Sept. 25, 1995. Uses "DATA.EQN" data.

b. title:  Parameter settings for this run:

c. listing: P01 =   33  P02 =    1  P03 =    1  P04 =   10
            P05 =    1  P06 =   10  P07 =    1  P08 =    1
            P09 =    2  P10 =    1  P11 =   12  P12 = 500
            :           :
            :           :
            P29 = 7.00  P30 =  .30  P31 =  .10  P32 = 1.00
            P33 = 1.00
  
```

**Fig. 6** Listing of parameter settings (in Auxiliary Output file)

## 2. Information about each person's links

If you set **P20 (auxfileout)** to a value higher than 2, you get information about the strengths and types of links of each individual in the network:

- a. A line of text saying "CHAIN -- LINK INFORMATION".
- b. A heading that indicates what follows and what the format is.
- c. The actual information.

Figure 7 shows the location of the 27 values in each group of three lines and an example for one individual in a demonstration network.

There are 27 values for *each person*. Figure 8 (on the next page) describes these values. Each person's information occupies three lines in the file, with 9 values per line.

- The first line has information about *reciprocated/confirmed* links (V1 - V9).





$V1 = \text{sum of strengths of outgoing parts of two-way links}$   
 $V2 = \text{number of two-way links with this person}$   
 $V3 = \text{mean strength of outgoing parts of two-way links} = V1/V2$   
 $V4 = \text{sum of strengths of incoming parts of two-way links}$   
 $V5 = \text{number of two-way links with this person} = V2$   
 $V6 = \text{mean strength of incoming parts of two-way links} = V4/V5$   
 $V7 = \text{sum of strength of both halves of two-way links} = V1+V4$   
 $V8 = \text{two times the number of two-way links} = 2*V2$   
 $V9 = \text{mean strength of all two-way links} = V7/V8$

$V10 = \text{sum of strengths of outgoing one-way links}$   
 $V11 = \text{number of outgoing one-way links}$   
 $V12 = \text{mean strength of outgoing one-way links} = V10/V11$   
 $V13 = \text{sum of strengths of incoming one-way links}$   
 $V14 = \text{number of incoming one-way links}$   
 $V15 = \text{mean strength of incoming one-way links} = V13/V14$   
 $V16 = \text{sum of strengths of all one-way links} = V10+V13$   
 $V17 = \text{total number of one-way links} = V11+V14$   
 $V18 = \text{mean strength of all one-way links} = V16/V17$

$V19 = \text{sum of strengths of all outgoing links} = V1+V10$   
 $V20 = \text{number of all outgoing links} = V2+V11$   
 $V21 = \text{mean strength of all outgoing links} = V19/V20$   
 $V22 = \text{sum of strengths of all incoming links} = V4+V13$   
 $V23 = \text{number of all incoming links} = V5+V14$   
 $V24 = \text{mean strength of all incoming links} = V22/V23$   
 $V25 = \text{sum of strengths of all links} = V19+V22$   
 $V26 = \text{total number of links} = V20+V23$   
 $V27 = \text{mean strength of all links} = V25/V26$

**Fig. 8** Contents of information-about-links segment



- The "A" is the node's "name" if names are used.

## 5. Listing of internal data structure

If you set **P20** (**auxfileout**) to a value higher than **3**, you get a complete "as is" listing of the internal data structure resulting from a run. The structure is divided into sections for (a) groups, (b) nodes, and (c) links. In each section there is one line per group, node, or link. If this option is chosen, the format of the information is described on the file itself. The contents are described briefly here. (If you need more information, contact the author at Simon Fraser University.)

### A. Group information

There are five values for each group:

- **GB**: the ID# of the last person in the group
- **GC**: this value is not used
- **GD**: the number of people in the group
- **GE**: this value not used
- **GP**: the ID# of the first member of the group

### B. Node information

There are five values for each node:

- **XB**: the ID # of the next person in this group
- **XC**: this value is not used
- **XD**: the group to which this node belongs/is attached
- **XE**: the role of this node (see Figure 11)
- **XP**: the location of this node's first link (0 ==> no links)



## MONITORING YOUR RUN

### Description

The program will show on the screen what it is doing as it progresses through the analysis. The amount of information you get is controlled by **P21 (monitorinfo)**. The higher the value you supply for this parameter, the more detailed the information will be. The highest value for **P21** is 5.



## 1. Parameters for describing your data

The following 12 parameters (**P01** through **P12**) are used to describe important features of your data to NEGOPY. These parameters specify how the data is structured, how it is to be read, and how it is to be interpreted. They also indicate how the data is to be cleaned as it is read in and pre-processed.

**P01 highestidnumber** <sup>4</sup>      **[high]** <sup>5</sup>      **Default=0 Min=1 Max=500**

(Highest ID Number)

Set this parameter to the highest ID number. Any links involving ID numbers higher than **P01** will be ignored by the program.

Ideally, ID numbers will start at **1** and go up to **N**, where **N** is the number of people. If numbers do not start at **1**, use **P02** to set the starting value. If there are gaps in the ID numbers, you are advised to run your data through the RENUMBER program that comes with NEGOPY.

**Note:** Do NOT set **P01** to a value higher than the highest ID number. Doing this will cause the program to identify spurious Isolates, each of which will have one of the ID numbers between the actual highest one and the value of **P01**.

**P02 lowestidnumber**      **[lowe]**      **Default=1 Min=1 Max=499**

(Lowest ID Number)

Set this parameter to the lowest ID number in your network. If the lowest ID number is **1**, you can leave **P02** at its default value.

**P03 Not Used**      --> Don't set this parameter. <--

**P04 namewidth**      **[name]**      **Default=0 Min=0 Max=20**

(Name Width)

You can associate an alphanumeric field with each person in the network. Although it is usually used to specify the names of the people in the network, "names" can be any kind of alphanumeric information. They can be up to 20 characters long. Set this parameter to the length you want to use.

---

<sup>4</sup> This parameter *must* always be set.

<sup>5</sup> This is the shortest abbreviation you may use to set this parameter.









**P15 linkinformation**                    **[link]**      **Default=1**    **Min=0**      **Max=4**

(Link Describe)

This is the first of the output control parameters. For **P15** through **P21**, the higher the value you specify, the more detailed information you get. If you set one of these parameters to **0**, you won't get any information from that part of the program.

**P15** controls the amount of information you get about links in the network. If you set it to **1**, you get a very simple list of the links of each node. If you set it to **4**, you get extensive lists, with descriptions of the two-way, outgoing one-way, and incoming one-way links. You also get tables of link strengths and comparisons of what each individual says about link strength with what the people linked to the individual say. Intermediate values for **P15** give you intermediate amounts of detail.

**P16 isolateinformation**            **[isol]**      **Default=1**    **Min=0**      **Max=4**

(Isolate describe)

This parameter controls the amount of information you get about isolates. (See the description for **P15**)

**P17 densityhistogram**            **[dens]**      **Default=1**    **Min=0**      **Max=1**

(Density Histogram)

If you leave this parameter at its default value of **1**, you will get a density histogram showing the initial detection of groups. This histogram is analyzed in the initial detection of groups. Seeing the histogram can be useful in understanding your data. If you set this parameter to **0**, no such histogram will be produced.

**P18 testinformation**                **[test]**      **Default=0**    **Min=0**      **Max=5**

(Describe tests)

This parameter is like **P15**, except that it controls the printing of information about the various tests the program conducts as it forms groups. Most of this information won't be of much value to you but you might find it interesting.







characteristics of your network. Leaving it set to the default value generally results in a finer breakdown of the network into smaller groups.

**Note:** *As this parameter plays a key role in the definition of group membership, you should not change it unless you have a theoretical or conceptual reason for doing so. You should report any changes you make when describing the results of your analysis.*

**P28 %withingroup**                      **[%wit]**    **Def.=50.01**    **Min=50.01**    **Max=100.0**

(% Within Group)

This parameter sets the percentage that is used in all the role definitions used in the analysis. The default value says that group members must have more than half of their linkage with other members of the same group. If you change the value to which **P28** is set, you are changing the definition of "group" and "liaison".

**Note:** *Do not change the value of this parameter unless you have a theoretical or conceptual reason for altering the definition of "group". **Do not set this parameter to a value below .501.** The program will not accept such values, because they may create internally inconsistent results.*

**P29 minimumsplitsize**                      **[mins]**    **Default=12**    **Min=5**    **Max=100**

(Minimum Split)

After it has found tentative groups, the program examines each one to see if it is one group or if it is "really" two or more groups stuck together. It does this by trying to split the groups apart. The smallest group it will try to split will have at least **P29** members. For more information on the effects of different values for this parameter, see "Performance and Sensitivity" (Chapter 8). The default value of **12** works well in most situations.

For small networks, the program automatically sets **P29** to a value of **7**, unless you set it to something else in the parameter setting file. If you set it to something else, the value you provide will override this automatic feature of the program.

**Note:** *If you set **P29** to a value higher than the default, it may inhibit the formal group-testing procedures and cause the program to produce unstable and invalid results. See "Performance and Sensitivity" (Chapter 8).*









individuals in the network, showing the role category to which the individual was assigned, the group with which the individual is affiliated (if there is one), and the individual's integrativeness score.

## A. Initialization and Preprocessing of Data

The output begins with information you use to confirm that the program has been properly instructed for the analysis of your data. A listing of the parameters and their settings is followed by a sample of some of the data as it is read in, so you can verify that your input format specification is correct. Finally, there are some statistics on the data that have been read in, showing how many lines of data were encountered, how many links there were, etc.

### Run Title/Header

The first page of output begins with a two-line header that tells you what version of the program you are using, followed by a line that says

"Your identification labels and control parameters appear below"

The next two lines on the page begin with "label a =" and "label b =", and show the two lines of information that you supplied at the beginning of your parameter specification file. These two lines will be printed from time to time throughout the analysis, and you can use them in order to describe the data, date, analytic options, etc. You should check this information in the printout, to make sure that it is your identification information that appears here, rather than the first line of parameter settings. An example of these headings is shown in Figure 1.

```
Negopy network analysis program: IBM Version 4.302
November 1995. (c) Copyright 1995, William D. Richards, Jr.

Your identification labels and control parameters appear below.

label a: "Demonstration/verification run for PC 4.302 Version"
label b: "Created November 26, 1995, by WDR."
```

**Fig. 1** Program version and label information



..... CONTROL PARAMETERS .....			
NUMBER	NAME	VALUE	COMMENT
1	(Highest ID )	249	**USER**
2	(Lowest ID Number )	1	DEFAULT
3	--> not used <--		
4	(Name width )	0	DEFAULT
5	(Reciprocation )	0	**USER**
6	(Unrecip Minimum )	0	DEFAULT
7	(Mean Stren Set )	1	**USER**
8	(Direction )	0	DEFAULT
9	--> not used <--		
10	(Multiple links )	1	DEFAULT
11	(Low Weight Limit )	5	**USER**
12	(High Weight Limit )	150	**USER**
13	(Check Link Stren )	3	DEFAULT
14	(Number Raw Print )	10	DEFAULT
15	(Link Describe )	2	**USER**
16	(Isolate Describe )	2	**USER**
17	(Density Histogram )	1	DEFAULT
18	(Describe Tests )	0	DEFAULT
19	(Group Describe )	2	**USER**
20	(File Output )	0	DEFAULT
21	(Monitor Level )	2	**USER**
22	(Print D-Matrix )	1	DEFAULT
23	(One Output File )	1	DEFAULT
24	(Number of Iteration )	4	DEFAULT
25	(Scan Radius )	200	DEFAULT
26	(Grp Sensitivity )	100	DEFAULT
27	(2 Links in Group )	0	DEFAULT
28	(% Within Group )	50.010	DEFAULT
29	(Minimum Split )	12.000	DEFAULT
30	(Split Std. Dev. )	0.300	DEFAULT
31	(Drop Split )	0.100	DEFAULT
32	(Transweight )	0.0	DEFAULT
33	(2-Step Weight )	2.0	DEFAULT

**Fig. 2** Parameter settings table



After the diagram you will see both the strength transformation equation the program is using and the format specification for your data file. It will look something like this:

Equation evaluated:  $\text{STRENGTH} = A * A * B * B$   
 Here is your input format: (I3,I4,8X,2F2.0)

## Listing of Segment of Raw and Processed Data

If you left **P14** (**nrawprint**) at its default value (4), or if you set it to any value other than 1, a listing of the first **P14** lines of your raw data begins at this point. The listing for each line of data begins with the notation

DATA LINE # xxx, AS READ IN. (RESPONDENT IS # xx)

followed by the ordinal number of that line in the datafile. (The program calls the first line read "#1", the second "#2", etc.)

Each link is then described with three lines of information, as shown in Figure 4. Here the first line of data contains a link from individual #1 to individual #2. The second line of data contains a link from #1 to #3.

In the data for this example were two strength weights for each link. The weights for the link from #1 to #2 were 3.0 and 4.0. The weights for the second link were 2.0 and 3.0. The lines marked "→" show the processed links obtained from this data. The strength transformation equation used with the example would have the effect of multiplying the two strength weights together and then squaring the result. Thus, there is a link from #1 to #2 with a strength of 144, and one from #1 to #3 with a strength of 36.

```

Data line 1      (Respondent is # 1)
As read in:
  -- from # 1 to # 2  weights: 3.0  4.0
  After processing:
  → -- From # 1 to # 2  strength= 144
-----
Data line 2      (Respondent is # 1)
As read in:
  -- from # 1 to # 3  weights: 2.0  3.0
  After processing:
  → -- From # 1 to # 3  strength= 36
-----

```

**Fig. 4** Display of segment of raw and processed data





connections? Is a reciprocated link one link or two? If the data contains three reports of a link from **A** to **B** (one for production-related contact, one for social contact, and one for contact about questions and changes), how many links are there between **A** and **B**? These issues can be resolved by distinguishing between *links* and *link components*.

A *link component* is one individual's report of a link with another individual. A *link* is a connection between a pair of individuals, *regardless of the number of individuals reporting the connection*. Therefore, if one person reports a connection with a second person, there will be one link component and one link. If the second person also reports a connection with the first, there will be **two** link components, but still only one link.

If there are three reports of a connection from **A** to **B** (each describing, perhaps, contacts involving different content areas or different communication media), there will be three link components, but still only one link. In this case the link from **A** to **B** is called *multi-stranded*.

"Multi-stranded" is close in meaning to, but not exactly the same as, "multiplex". A *multiplex* tie between a pair of persons means that the persons relate to one another in different ways. They may have a formal relation to one another at work; they may be close friends; and they may be members of the same hockey team. To say that a link is multi-stranded is to say that the connection between the pair of individuals is described more than once in the data. This may be due to error (the same information was entered more than once more than once in the file like like this). Or it may be due to the fact that the measurement and coding procedures made it appropriate or necessary, in which case these multiple links would not be errors, and should not be treated as errors.

If your measurement procedure recognizes multiplexity of ties and provides a way of measuring different kinds of relationships, you would expect to have multi-stranded links in your data. Some examples of situations that might require multi-stranded links are described here:

- The data is obtained by analyzing a videotape of the events that take place in a closed environment. Each interaction is coded as it takes place. The information associated with each interaction includes the ID numbers of the participants, the time at which the interaction began, the time at which the interaction ended, etc.

This will produce a dataset in which a pair of individuals may have dozens of descriptions of interactions. For analysis, the duration of each interaction might be calculated, and the total amount of time spent in interaction might be used as an index of the



## Processing ID numbers

NEGOPY produces a summary that includes the kind of information shown in Figure 5. The components of this listing are explained here:

Number of lines processed = 13

This specifies the number of lines of data that were processed. In this case, 113 lines of data were processed. This number ordinarily will correspond with the number of lines in your data file.

No. of lines rejected: bad resp. ID# = 0

This specifies the number of lines of data that were rejected because the respondent's ID number was too large or too low. In the example, no lines were rejected. If you have a large number of rejected lines, it could be due to an improper format specification for your data, because of data entry errors, because the value of **P01** or **P02** were incorrectly set, or because something happened to your data file that altered the column locations.

No. of lines rejected because of bad contact ID# = 0

This specifies the number of lines that were rejected because the ID number of the other individual was too high or too low. Problems here can be caused by the same factors as in the above paragraph. It is more likely that there will be errors in data entry in this category than for respondent ID numbers, simply because most data preparation systems allow the grouping of links by the respondents' ID numbers, while this is not possible for the other ID numbers. The ID numbers will have been checked to see that they are not greater than **P01** [**highestidnumber**] and not less than **P02** [**lowestidnumber**]. They will also have been checked to see that there are no *recursive* links, in which an individual reports a link with him or herself.

## Testing Link Strength

The program tests the strength of links as the links are read in and the strength is calculated. Actually, it is more correct to say that the program tests the strength of *link components*, rather than of links. If a link component has a strength greater than the value of **P12** [**strongestlink**] or lower than the value of **P11** [**weakestlink**], that link component will be eliminated from the analysis, even though it may be part of a reciprocated link, where the strength of the *link* is not known until the two halves of the



no. of lines processed	=	113
no. of lines rejected: bad resp. ID#	=	0
no. of lines rejected: bad contact ID #	=	0
no. of lines rejected: strength 0.0 or less	=	1
no. of lines rejected: divide by zero or imaginary root	=	0
Link strength has been tested.		
no. of lines dropped: strength too low	=	41
no. of lines dropped: strength too high	=	0
no. of good lines processed	=	71
** While summation of multiple links or processing	**	
** of unconfirmed links may change the strengths of	**	
** links, link strength will not be tested again.	**	
Multi-stranded links have been found.		
They have been converted to single links		
by ignoring 2nd and subsequent strands.		
no. of extra strands found	=	8
no. of confirmed links found	=	13
no. of unconfirmed links found	=	37
no. of unconfirmed links dropped	=	8
no. of markers added for unconfirmed links	=	29
no. of link components for analysis	=	55

Summary of data read in a run

Where Figure 5 shows that 8 extra strands were found, it means that there were a number of links that had extra components or "strands". The number of "extra strands" is eight.

Multi-stranded links have been found.

They have been converted to single links  
by ignoring 2nd and subsequent strands.

no. of extra strands found = 8

These lines in the display show that the multi-stranded links have been converted to single links by deleting all but the first strand for each link (this assumes multiple links are errors). This is one of two options available by means of **P10**. The other option is to add together the strengths of second and subsequent links between any pair of individuals. The number of link elements that remain after processing multiple-stranded links will reflect whatever action was taken after these links were identified.



## B. The Links of Individuals

The next section of the output focusses on the links of the individuals in the network. It begins with a listing of the links of the members of your network that includes a variety of descriptive and summary information. There is also a histogram showing how many links were encountered at each level of strength.

Several kinds of information are provided in this section, depending on how you set **P15** [`linkdescribe`]. Examples are shown of the level of detail you receive for different settings.

-----						
Links of node #	7	(G. Barnett.)	out:	in	[mean]	discr.
with node #	25	(E. Rogers.)	40:	50	[ 40]	-10
from node #	27	(B. Wellman)	50:			
to node #	32	(R. Wigand)	50:			
-----						
Links of node #	8	(R. Brieger)	out:	in	[mean]	discr.
from node #	21	(J. Moreno)	36:			
to node #	24	(L. Freeman)	32:			
with node #	29	(J. Coleman)	24:	20	[ 22]	2
-----						

**Fig. 6** Link list for **P15** [`linkdescribe`] = **2** (medium detail)

Figure 6 shows the results of setting **P15** to a value of **2**. This medium level of detail is appropriate for many users. The main body of this information is the display of the links of all the members of the network. The links of each individual are shown separately and arranged so that any particular link is easy to find. Figure 6 shows the links of individual #7 and individual #8. Several pieces of information are provided with each link:

- If the link is confirmed (two-way), the word "**with**" will appear before the word "**node**". Otherwise, either the word "**to**" or the word "**from**" will appear, depending on whether this link was reported by the individual whose links are being listed or not. If you look at the links of G. Barnett (#7) in the example above, you see a link "**to** node #32 (R. Wigand). This means that Barnett reported a contact with Wigand, who did not confirm that report. (From Barnett's perspective, the link goes *to* Wigand). You also see a link "**from** node #27 (B. Wellman). This





Figure 7 shows the information you get if you set **P15 [linkdescribe]** to the maximum value of **4**. The display is the same as it was in Figure 6, except that a "Discrepancy table" and a "Strength table" are included.

Links of node # 14	(	E Katz)	out:	in	[mean]	discr.
with node # 4	(	J P Boyd)	50:	50	[ 50]	0
to node # 6	(	F Harary)	20:			
to node # 9	(	O Frank)	50:			
with node # 12	(	J Braun)	36:	42	[ 36]	-6
with node # 15	(	Kadushin)	24:	18	[ 36]	6
to node # 17	(	R Burt)	35:			
to node # 18	(	T Valente)	25:			
from node # 21	(	J More)	30:			
with node # 23	(	A Wolfe)	15:	35	[ 15]	-20
-----						
Discrepancy table						
			sum	mean		
	actual		-20.	-5.00		
	absolute		32.	8.00		
-----						
strength table						
	out		in		total	
2-way	s=125 n= 4 m= 31.25		s=145 n= 4 m= 36.25		s=270 n= 8 m= 33.75	
1-way	s=130 n= 4 m= 32.50		s= 30 n= 1 m= 30.00		s=160 n= 5 m= 32.00	
total	s=255 n= 8 m= 31.87		s=175 n= 5 m= 35.00		s=430 n= 13 m= 33.08	
-----						

**Fig. 7** Link list display for **P15 [linkdescribe] = 4** (high detail)

The discrepancy table gives you an analysis of the discrepancy scores for each individual. The numbers in the first row of this table refer to the actual discrepancy scores for the individual, while the numbers in the second row refer to the absolute values of the discrepancy scores.

If an individual has consistently positive (or consistently negative) discrepancy scores, the values in the "actual" row will be very close in size to those in the "absolute" row. If an individual has some positive discrepancy scores and some negative ones, the size of the values in the "actual" row will be much smaller than the size of the values in the "absolute" row, because some of the positive scores will cancel out the negative ones when



of the five has contact with all four of the others, there will be 10 links. If all link report are confirmed, there will be 10 two-way links and 20 link elements. If one of the five members were absent from the data collection, but each of the other four reported contacts with one another and with the fifth, there would still be 10 links but only 6 of these 10 links (60%) would be two-way.

```

*****
* Strength distribution analysis *
* lowest legal value = 120.0 *
* highest legal value = 100.0 *
* max. observed value = 50.0 *
* range is 38 units *
*****

```

**Fig. 8** Strength distribution summary

After the Reciprocation/Confirmation Analysis table you will see another table, called Strength Distribution Analysis. This table, shown in Figure 9, indicates the range of strength values that were observed in your data.

```

*****
* Reciprocation/Confirmation analysis *
* two-way oneway total *
* number 13. 29. 42. *
* percent 30.95 69.05 *
*****

```

**Fig. 9** Reciprocation/Confirmation table

Following this table is the Strength Distribution Histogram, an example of which is shown in Figure 10. The purpose of this histogram is to give you an indication of the number of links at each level of strength. This will be useful as you try to understand what is happening with your data. If you notice a very small number of extremely strong links, you might want to verify that these links are legitimate. If you see that all the links have the same strengths, or that there are no links with strengths in a range where you have



## C. Information about Group Detection

This section contains information about the group detection process and about any groups the program finds. The section begins with a measure of the amount of structuring or organization in the network. This index gives you an objective measure of the extent to which the network as a whole is structurally differentiated. Next is the display of the density histogram which is used in the initial detection of groups. This is followed by a listing of the members of each group and a distance matrix for each group in the final analysis.

### Structural Differentiation

Figure 11 shows the information the program uses in its calculation of the amount of structural differentiation. The last line shows the measure of overall structure, which ordinarily runs from zero to one. A value of zero indicates the network is no more differentiated than would be expected in a "random" network.

```

*****
*                               *
*      Structure Calculations    *
*                               *
* Observed no. of Nodes      = 23. *
* Observed no. of Links     = 37. *
* Observed System Density   = 0.146 *
*                               *
* Expected no. of Cycles    = 5.056 *
* Maximum no. of Cycles    = 33.00 *
* Observed no. of Cycles   = 11.00 *
* STRUCTURE (S)            = 0.2127 *
*                               *
*****

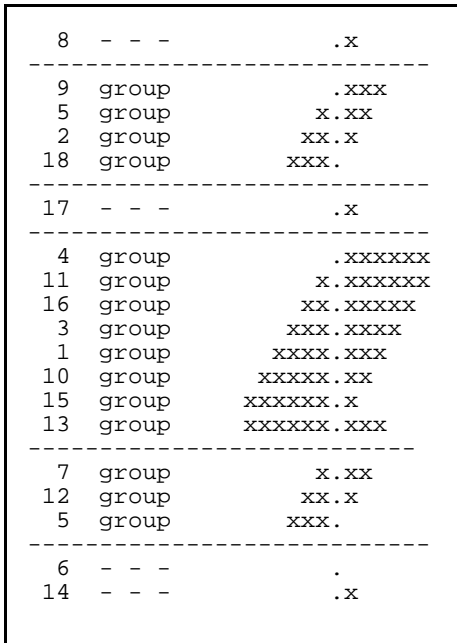
```

**Fig. 11** Measure of structural differentiation

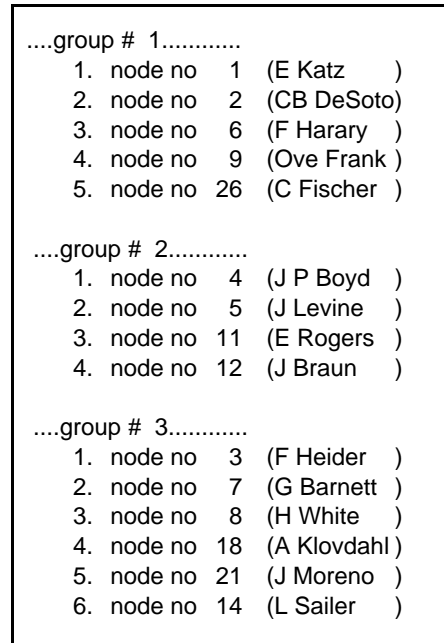
This measure is based on the amount of what might be called "local interconnectiveness." If a network is highly structurally differentiated, there will be a number of relatively small groups. Almost all of the links of the members of each group will be with other members of the same group. Most of the members of each group will have links with most of the other members of their group. There will be few links between members of different groups.



along the left column are ID numbers. Three groups were tentatively identified in the example. The first contains the individuals with ID numbers 9, 5, 2, and 18. The second contains 4, 11, 16, 3, 1, 10, 15, and 13. The dashed lines across the histogram signify the beginning of new groups (the program begins its scan of the histogram from the top and works down toward the bottom), so the first group begins with ID # 9 and the next begins with ID # 4.



**Fig. 12** A simple density histogram



**Fig. 13** List of groups and members

Notice that there are a number of rows that do not have the word "group" next to the ID number. These individuals (8, 17, 6, 14, & 19) have not been assigned to the tentative groups. If you look on the right side of the histogram for the individuals not assigned to groups, such as 8 and 17, you will see that there are few or no **X**'s on these lines. Chapter 7 has more information about the use of the density histogram.

Following the density histogram and the information associated with it is a simple listing of the groups and their members, such as the one shown in Figure 13.

*Note that this is not necessarily the final group structure of the network. A number of tests still must be performed on this tentative solution.*





If you are using only confirmed/reciprocated links, or if you have selected the non-directed option by leaving **P08 [direction]** set to its default value of **0**, your distance matrices will all be symmetrical. This means that the upper right half will be a mirror image of the lower left half.

If you selected the directed analysis option by setting **P08 [direction]** to **1**, you will receive two distance matrices for each group that is identified. The first will be symmetrical, while the second might not (provided that you have not rejected all unconfirmed links by setting **P05 [reciprocate]** to zero). In the second matrix, you may see some zero entries off of the diagonal. If the entry in row *i*, column *j* is a zero, there is no path from individual *i* to individual *j*. *Note that this interpretation will be valid only if your data represent directed relationships.*

### Analysis of the Distance Matrix

Immediately following the distance matrix for a group are two tables headed "Analysis of Distance Matrix". An example is shown in Figure 15. The first table has one row for each member of the group. In this table, the rows are arranged so that the one for the individual with the lowest ID number comes first, and the one for the individual with the highest ID number comes last. There are four values for each member of the group in this table. The first is the sum of all the entries in the individual's column in the distance matrix. The second is the mean of these column entries. The *column mean* tells the mean number of steps it takes to get from other individuals in the group to this individual. The third and fourth entries give similar information, but based on row entries instead of column entries.

Analysis of distance matrix									
node #	col total	col mean	row total	row mean	node #	row mean	dist ratio	std. distance	
2	5	1.25	5	1.25	5	1.00	0.77	-1.434	
3	6	1.50	6	1.50	7	1.25	0.96	-0.239	
5	4	1.00	4	1.00	2	1.25	0.96	-0.239	
6	6	1.50	6	1.50	6	1.50	1.15	0.956	
7	5	1.25	5	1.25	3	1.50	1.15	0.956	
group col mean = 1.30					std. dev. = 0.2092				

Fig. 15 Analysis of Distance Matrix



The third value for each individual in this table is the *standard distance*. This is a z-score; it is calculated by dividing the difference between the individual's mean and the group mean by the group standard deviation. It is an alternative way of expressing how close each member of the group is to the others, relative to how close the others are. The standard distance is probably more useful in large groups than it is in smaller ones, mostly because the values will be more stable in larger groups.

## E. Individuals in Groups: intensive analysis

The last major section of the output is comprised of a detailed analysis of the links of the members of each group. Depending on how much detail you asked for with **P19** [**groupdescribe**], you will get one large table that summarizes the contacts of the members of the group, additional tables summarizing the contacts of each member of the group, and detailed listings of the links of each of the members.

### Link Analysis Tables

These are the tables that summarize the links, either of the entire group, or of an individual member of a group. Each set of Link Analysis Tables is composed of two, four, or six sub-tables. Both groups and individual members have sub-tables that show information about the *number of links* and the *amount of interaction*. Groups also have information about the *mean zone overlap* of links and about the *mean strengths* of links. In addition, if members of the group have contact with liaisons or members of other groups, there are two additional tables that describe the *number of links* and *amount of interaction* with *individuals who are not members of the group*. Each type of table is described separately in the paragraphs that follow.

A few notes on how links are counted in these tables are in order because the counting procedure here differs somewhat from the one described earlier. *Please note that the method described here differs from the method described in earlier versions of the program manual.*<sup>6</sup>

---

<sup>6</sup> Earlier versions described each link from the perspectives of the two individuals connected by the link. A one-way within-group link thus counted once as an outgoing link and once as an incoming link, since from the first member's perspective it went *to* the second member, while from the second member's perspective it came *from* the first member. The current version

(continued...)



1. Number of outside-group links	did not exist prior to 1989
2. Amount of outside-group interaction	did not exist prior to 1989
3. Number of links	different from pre-1989 version
4. Amount of interaction	different from pre-1989 version
5. Mean zone overlap	same as pre-1989 version
6. Mean strength	same as pre-1989 version

### Number of outside-group links

Figure 16 shows a typical example of the "Number of Outside-Group Links" table from a group's summary. In the example, there are rows for the links members of this group have with members of other groups, with direct liaisons, and with indirect liaisons. In many cases there may be only one or two rows actually present in the table -- if there are no links in one or more of the categories, no row is printed for that category. There are columns for two-way links and outgoing & incoming (one-way) links.

The purpose of this table is to show the number and type of connections the group has with the rest of the network.

This table shows that there are 71 links between members of the group and non-members. The Totals column shows that 49 of these (69.01%) are with members of other groups, while the other 18 (25.35%) are with direct liaisons and four more (5.63%) are with indirect liaisons.



The table shown in Figure 17 shows a slightly different picture than the one in Figure 16. For example, the previous table showed that just over 40% of all links were one-way. But here we see that these links account for almost 50% of the interaction crossing the group boundary.

In the tables shown in Figures 16 and 17, one-way links which originate in the group are kept separate from those which originate outside the group. The former appear in the "outgoing" column and the latter in the "incoming" column. This does not work for one-way links between members of the group: a one-way within-group link originates in the group, so it appears in the "outgoing" column, but it also terminates in the group, so it appears in the "incoming" column, resulting in double-counting. But it is only the within group one-way links for which this is the case. This problem is illustrated in Figure 18, which was produced by an earlier version of the program. The current version of the program avoids this problem by combining the "outgoing" and "incoming" columns into a single "one-way" column, as shown in Figure 19. Compare these two tables, and take special note of the percentages in the totals columns.

	Number of links			total	percent
	two-way	outgoing	incoming		
Within Group	43.	71.	71.	175.	68.36
Between Group	24.	12.	13.	49.	19.14
Liaison Dir	15.	2.	1.	18.	7.03
Liaison Indir	3.	1.	0.	4.	1.17
total	85.	86.	85.	256.	
percent	33.20	33.59	33.20		100.00

**Fig. 18** Number of links -- *old method*

In the table shown above, the "175" for the Within Group total is incorrect because it double-counts the outgoing and incoming (one-way) links. The correct total number of Within Group links should be  $43 + 71 = 114$ , which is the number shown in the "undirected" total column of the Within Group row.





Note that one-way links (10.83) appear to be stronger than two-way links (7.51). This result, as well as those of other tables involving strength comparisons, must be carefully interpreted in light of the fact that different minimum strengths can be set for one-way and two-way links. For the run that produced the results in the tables in Figures 17 - 21, the minimum strength of two-way links was set to 4.0. The minimum strength of one-way links was set to 9.0. These operations were carried out by setting the parameters **weakestlink** and **unrecipmin** to 4.0 and 9.0. Under these conditions, one would expect the one-way links to have a mean strength higher than that of the two-way links.

## Zone Overlaps

The next table, shown in Figure 21, is headed "Mean Zone Overlap". An individual's "1st-order zone" is the set of nodes with whom the individual is linked. The zones of two individuals will overlap to the extent to which they both have links with the same other nodes -- that is, to the extent to which members of one individual's zone are also members of the second individual's zone. The size of the overlap of the zones of two individuals is thus the number of two-step links connecting the two individuals. Within-group links tend to have larger zone overlaps because it is likely that the members of a group share contacts with other members. Links with liaisons tend to have the lowest zone overlaps, as they connect individuals not otherwise connected. The group whose table is shown in Figure 21 has a rather high overlap for two-way between-group links, which reflects the fact that the relatively large groups in this network were connected by a large number of between-group links. In networks where the groups are smaller and more clearly defined, with few between-group links, the zone overlaps for between-group links are almost as low as the overlaps for links with liaisons. In general, the overall amount of zone overlap in a network is highly correlated to the Structure Index which the program computes and prints near the end of the listing of the links of individuals.



value. This is not an error or a bug -- it happens because links are not counted the same when the directed option is selected, since selecting the directed option assumes a different definition of "link".

## Links going outside the group

After the Link Analysis Tables for each group, there is a description of all links with liaisons and links with members of other groups. These lists show who the link is with, how strong it is, whether the other individual is a liaison or a group member, and which group the individual belongs to. If you set **P19 [groupinfo]** and **P15 [linkinfo]** to maximum values, you will see an analysis of all links going outside the group. It shows the amount of linkage and number of links directed to and from members of the group, and can be used to determine whether the group as a whole is a net sender or receiver of information, as well as where the information goes to or comes from.

## Final Summary Tables

There are two more tables that appear after all information about links with group members. An example of the first one is shown in Figure 23. You should examine this table carefully and pay special attention to the number of isolates. There are three things that can cause there to be a large number of isolates:

- 1) You may have set one of the parameters that tell the program the highest and lowest legitimate ID numbers in your network incorrectly. You may have set **P01 [highestid]** to a value higher than the number of individuals in your network. If you do this, all the unused ID numbers between **1** and the value to which you set **P01** will be classified as Isolates Type 1. Alternatively, if the lowest ID number in your network is not **1** and you forget to set **P02 [lowestid]**, all the unused ID numbers between **1** and the actual lowest ID number will be classified as Isolates Type 1.



number of the group to which he belongs or is connected to, his integrativeness score, and his name (if you used a Namelist file). This table is a handy summary of the entire network's structure. It has sometimes been used as input to a general purpose statistical program. A condensed version of this table is shown in Figure 24.

The numbers in the last column of the final table are integrativeness scores, but they have all been multiplied by 1000, so you have to adjust the decimal points. A score of "90" in the table is actually "0. 09"

<u>ID#</u>	<u>classification</u>	<u>group</u>	<u>intgrt</u>
1	Group Member	1	90
2	Group Member	1	267
3	Tree Node	0	0
4	Group Member	3	15
5	Group Member	2	186
6	Group Member	2	231
7	Tree Node	0	0
8	Group Member	3	75
9	Group Member	3	363
10	Group Member	1	1000
11	Group Member	1	667
12	Group Member	2	0
13	Tree Node	0	0
14	Group Member	2	425
15	Group Member	1	267
16	Isolate (T2)	0	0
17	Group Member	4	62
18	Group Member	3	750
19	Group Member	3	500
20	Group Member	4	125
21	Group Member	2	333
		:	
		:	

**Fig. 24** The Last Table



## Preparation of Data File

If you have not already created the data file, you can probably save some time and effort by following these tips:

- **ID Numbers** -- Assign ID numbers to individuals in your network so they start at **1** and go up to **N**, where **N** is the number of individuals in the network. Do not leave any gaps in the numbering of nodes. **This is important for consistent and reliable results.** If your ID numbers do not run from **1** to **N**, an easy way to fix this is to run your data through the RENUMBER program supplied with NEGOPY.
- While it is not necessary to assign any particular number to any particular individual, it may be helpful if you do the assignment in some meaningful way. An alphabetical assignment will make it easy for you to find the names of individuals on the list, but in some situations it is better to give all the members of organizational units contiguous numbers. This will make it easier for you to work with the organizational structure. While you are preparing the assignment of ID numbers, you might as well create the listing of information that you will use as a Namelist. (see Chapter 4.)
- If you have not coded your data yet, now is the best time to review Chapter 3 on links and link strengths. Ideally your strength indicators will be scaled as ratio-level variables. The stronger the strength of a link, the higher the value of the strength indicator, and the value of the indicator should increase in direct proportion to the strength of the link. This kind of scaling will reduce the need for complex strength transformations, make your analysis easier to set up, make your results easier to interpret, and increase their validity.
- Include only one link per line of data in your data file. While this may result in a longer file, it will save a lot of time and effort in correcting or modifying the file, should that be necessary at a later time. ***The program will not accept more than one link per line of data.***





"talk with" (if you talk with me I must also talk with you...), you cannot be certain that this is how the question is interpreted and handled by the people who give you the answers you will use for data. The conclusions you draw about this will have some bearing on how you deal with unconfirmed/ unreciprocated links.

Decide whether you want a **directed** analysis. If your data can be treated as directed (it can if the relation upon which your data is based is **not** symmetrical), you may wish a second set of calculations where it is appropriate. These calculations are most useful when looking at the links of individuals (where you might want to compare what an individual says about her interactions with what other people say about them -- does person *A* give more information than she receives?) or at certain aspects of group structures.

Set **P08 (direction)** to **1** if you want a directed analysis. If you do not want this extra analysis or if your data are not directed, leave **P08** set to its default value, which is **0**.

Selecting the directed option will make the program produce two distance matrices for each group it finds. The first matrix for each group ignores the direction of contacts. The second one does not. The second matrix will let you see, for example, which members of the group are potential sources of information for most of the others and which members receive information from most of the others. In the directed matrix, it is possible for there to be no way of getting from some members to other members.

### 3. Confirmed & Unconfirmed links

So that the placement of individuals into groups and other role categories is reliable and consistent, individuals must "look the same" (i.e. have the same role category assignment) from all possible perspectives in the data structure. That is, if individual *A* looks like a member of Group 2 according to an analysis of his links, he must also look like a member of Group 2 according to an analysis of the links of other individuals (some of whom may have links with *A*). Since group membership is based on patterns of linkage, it is necessary to have a consistent description of an individual's links with others in order to determine whether or not the individual is a member of a group. In other words, it is necessary to know unequivocally whether or not individual *A* is linked to *B* and how strong that link is. This means that, at least for role assignment, if *A* is linked to *B*, ***B* must also be linked to *A***. Furthermore, the strength of the link from *A* to *B* must be the same as the strength of the link from *B*



one -- add markers for the missing halves. You should know why you are selecting the option you choose. (See the section headed 'Cautions' later in this chapter.)

- a) If you want to reject *all* one-way links, set **P05 (reciprocate)** to **0**. You should also leave **P06 (unrecipminimum)** at its default value of **0**.
- b) If you want to reject only the *weak* one-way links, leave **P05 (reciprocate)** at its default value of **1**. Determine the strength of the weakest one-way link you want to keep, and set **P06 (unrecipminimum)** to this value. Markers for the "missing halves" of one-way links with strengths of **P06** or higher will be added.
- c) If you want to force reciprocation/confirmation by adding markers for the missing halves of all one-way links, leave both **P05** and **P06** at their respective default values of **1** and **0**.

#### 4. Multi-stranded Links

Sometimes there will be more than one link from one node to another. This could be due to dataentry error, or because you are combining data from different kinds of networks. It could also happen if your are using a chronological record of activity, where one link is recorded for each interaction. Cases in which a pair of individuals are connected by more than one link are called *multi-stranded links*.

The program must reduce all multi-stranded links to single links before beginning the group detection and analysis process. There are two approaches available to accomplish this goal:

- a) Ignore all but the first link between any pair of individuals.  
If there *should* be only one link between any pair of individuals (i.e. if multi-stranded links are due to data entry error), set **P10 (multilinks)** to **0**. Once the program has encountered a link between a pair of individuals, it will ignore any other links between the pair.
- b) Add all the strands of multi-stranded links together into single strands.  
If you want the multiple links to be added together, leave **P10** set to its default value of **1**.



2. Determine the highest and lowest legitimate link strengths (after processing with the transformation equation).
  - Set **P11 (weakestlink)** and **P12 (strongestlink)** to the lowest and highest legitimate values.
  - Note that when setting these "lows" and "highs" you should consider the impact of summing the two halves of reciprocated links if **P07 (meanstrength)** was set to **2**, or of adding together multiple links if **P10 (multilinks)** was set to **1**.
  - **NOTE:** The strengths of links are not tested until after multiple links have been combined, links have been tested for confirmation/ reciprocation, and all other processing of links according to **P05**, **P06**, and **P07** has been completed.

### C. Controlling the Output

Seven parameters (**P14** to **P20**) are used to control the amount of output you get from various parts of the program. In general, the higher the value you assign to a parameter, the more information that part of the program gives you. Some of these parameters should be set for each run, while others will rarely (if ever) need to be changed from their default values.

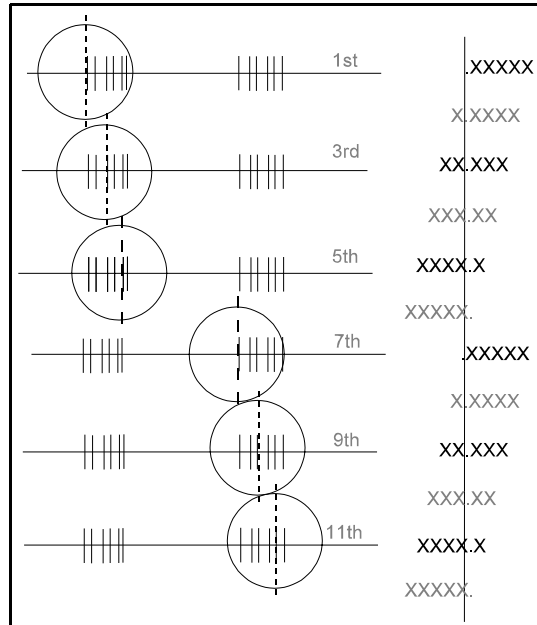
*The values you give to these parameters have no impact on the assignment of individuals to groups.*

- You should probably leave **P14 (nrawprint)**, **P17 (densityhist)**, **P18 (test-describe)**, and **P20 (auxfileoutput)** alone for most runs. The only time you would want to alter any of these from their default values is when you have a special requirement of some kind, where the extra information will be of some use.
- **P15 (linkdescribe)**, **P16 (isolatedescribe)**, and **P19 (groupdescribe)** will be the main parameters you will be using in this section. They control the output produced by the major parts of the program. All three of these parameters have the same range of permissible values, and all three are set to a default value of **1**. For your first runs on any set of data, the default settings are probably okay. The only exception to this is when you have a very large network with very large numbers of links. In this case, you might want to set **P15** to **0** for the first run, as lists of links can easily take up a few hundred pages of output in a large network.
- **P20 (auxfileoutput)** controls the preparation of a file that includes, among other



of the means visible through the window are on the right side. Below this one is the window centered on the third mean. Now two means are visible on the left side and three on the right. As the window moves through a block of means, the ones initially visible on the right side become visible on the left side. As the window moves further, the means drop out of view on the left side as new ones come into view on the right.

Whenever there are links between members of different groups, or whenever there are links with liaisons, the breaks between groups in the histogram will be more difficult to see clearly. Also, other factors may affect the clarity of the network's structure. For example, if you combine the data from two different content or functional categories, you are likely to obscure much of the structural differentiation because the networks will not be identical, and some of what is present in one is likely to overlap with what is not present in the other. The result is that the pattern in the density histogram will resemble one big blob, rather than the clear clusters shown in the examples above. In some cases the program will detect only one large group rather than a number of smaller ones.



**Fig. 2** The scanning window in operation

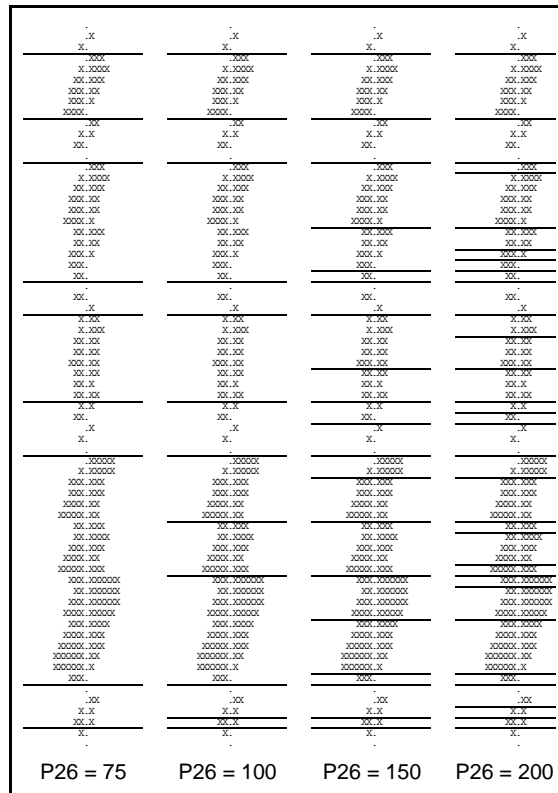
There are two parameters you can use to adjust the way the program constructs and interprets the density histogram: **P25 (scanradius)** and **P26 (sensitivity)**.

**P25** controls the width of the "window" used to construct the density histogram. The wider the window, the more nodes are visible through it, and the larger the clumps the program is likely to detect. If the window is large enough, the program will see only one big clump, and it may be unable to separate the subgroups, if there any. If you make the window too small, the program will not see enough nodes through the window to make any groups at all. The program is usually not very sensitive to small variations in the size of the window -- variations of as much as 50 to 80 percent can have little impact on the results, although in some unusual data situations, smaller variations can have larger





**NOTE** -- the groups identified by the use of the density histogram and the window are only a first approximation of the group structure. You may get the same final results with a small window as you get with a larger one, even though the program initially finds a different number of groups by analyzing the density histogram. This is because of the logical/ mathematical testing the program conducts in later stages of analysis.



**Fig. 4** Effect of changing **P26**

**P26 (sensitivity)** controls the sensitivity of the group detection part of the histogram analysis procedure. If you raise this parameter to a higher value, the scanning process becomes more sensitive, and more tentative groups will be identified. If you can see several clumps on the density histogram and feel the program has not drawn lines between the clumps where it should have, you might try raising the value of **P26** by 20 to 30 percent. Even if this does result in more tentative groups, it does not guarantee that these groups will all make it through the formal testing that follows. It does,



instead of 2.) See the section on "Strength Transformations" in Chapter 4 for an in-depth discussion of exponentiation in both versions of the program. As in the case of deleting weak links, you should use this method only if you have a theoretical or conceptual justification for altering the scaling of link strengths.

- If your conceptualization of "group" emphasizes the tight integration of members' communication networks, in the sense that any two members of a group are likely both to be directly connected to one another and to have many contacts in common, you might raise the value of **P32 (transweight)**. When this parameter is left at its default value, **1.0**, any two links of the same strength have the same weight in determining the role classification of an individual. If you set **P32** to a higher value, transitive links will have heavier weights than intransitive links of the same strength. (A transitive link is one that connects two individuals who both have contacts to the same other people. For example, if **A** and **B** both have links with **D** and **E**, the link between **A** and **B** will be transitive.) Because very densely interconnected areas of the network will have a higher proportion of transitive links, raising **P32** will increase the ability of the program to identify these areas as groups.

*If you use transitivity weighting in this manner, you should explain what you did in your research report.*

- If your data describes links involving several content or function areas, it would be better to separate the content/function areas and treat each one as a separate network, which will do much less violence to the data. I have seen many cases in which a combination of multiple content areas produced no groups, where the same data produced several very clear groups when the content areas were run separately. This underscores the need to carefully design your approach to measurement -- i.e. to have a clear conceptualization of the relationship you are studying and an understanding of the implications of different measurement and scaling methods.
- If you do not have multiple content/function areas, but you feel there are clearly distinguishable areas in the density histogram, you might consider adjusting the parameters that determine how the group detection process proceeds. There are two approaches you can take here:
  1. You can increase the sensitivity of the routine that detects breaks between groups. Do this by raising the value of **P26 (sensitivity)** by 25 to 50



of the distance matrix. If the standard deviation is higher than the value of **P30 (splitsd)**, the program attempts to split the group apart. The process used to do this is fairly expensive, computationally speaking, so the program tries to avoid going through this process unless it seems like it would be useful.

If the group is too small, the splitting process may make the group fall apart, but subsequent testing may cause it to be put right back together again. The size of the smallest group the program will test is controlled by **P29 (minimumsplit)**.

If the group is large enough and the standard deviation is high enough to trigger the initiation of the splitting process, a percentage of the group's most central members is temporarily removed. The size of this removed segment is controlled by **P31 (dropsplit)**.

In very small networks, it may be necessary to reduce the value of **P29** from the default value of **12** to something like **7** or **8**. If this is done, it will be necessary to raise the value of **P31** from its default **0.1** to something like **0.13** or **0.15**. You should set it to a value that is large enough so that when you multiply it by the size of the smallest group you want to test, the result is greater than or equal to 1.0. If you fail to do this, the program will try to take 0.1 of the group, and that may be less than one member. In that case, it will remove no members and the test will not be carried out.

To summarize, if you want the program to test groups that have fewer than 10 members, you have to raise the value of **P31**. If you want it to test groups that have fewer than 12 members, you have to lower the value of **P29**. If you want it to try to split groups that have less variability than normal, you have to lower the value of **P30**. See "Performance and Sensitivity" for further discussion.

## Using the Distance Matrix

The distance matrix is also utilized in the determination of group structures. Like the density histogram, the distance matrix can be used to determine whether or not you should adjust some parameters and submit another run. There are two conditions that you should look for. Each requires its own approach:

- If you have a small number of very large groups and the distance matrices for these groups are filled with mostly 1's and a few 2's, this means that the individuals in your network are very tightly connected together. Each individual has a large number of connections to others in the network, or else there are a few



that would be liaisons if there is more than one group) and then test to see if it falls apart. If the group is still connected, the removed members are automatically put back in.

In some cases (e.g. if your groups have fewer than ten members each) it may be necessary to remove more than ten percent to remove even one member. In other cases (e.g. when there are high levels of density) it may be necessary to remove as much as fifteen to twenty percent to make the group break up, and in some cases, you have to remove 30 or 40 percent of the group's members. I would suggest that this would indicate that you have one big group and that you should not continue trying to break it apart.

Note that altering this parameter implies that you are altering your definition of what a group is. If you choose this approach, you should include an explanation of what you did and why you think it is justifiable in your research report.

## **Running NEGOPY**

This section describes how you associate files with the program as you run it. Once you have prepared your data file, your parameter setting file, and your namelist file (if you are using one), you are ready to submit a run to the computer. You need a minimum of four files (your data, your parameter settings, a file for the output, and the default parameter definition file). You can also use a namelist file, and an auxiliary output file.

To make the program go, simply type its name. The program will ask you for the name of each file as it needs it. Type in the names of the files as it asks for them.

If the files are in a directory or drive other than the one you are running the program from, you will have to include the path as part of the file name. For example, say you are running the program from `c:\negopy\` and your data (`link.one`), parameter settings (`parm.one`), and namelist (`names.one`) files are in the subdirectory `c:\negopy\orgcom\`. When the program asks for the parameter setting file, you should respond with `"orgcom\parm. one"`. When the program asks for the data file, you should respond with `"orgcom\link. one"`.

The filename, including the path, may not exceed 24 characters. If having to include the path makes it necessary to use more than 24 characters, you should either move the files to a different subdirectory so the path will be shorter, or you should run the program from the subdirectory in which your files reside.





## Cautions

NEGOPY is a lot like many statistical procedures, in the sense that it makes a number of assumptions about your data and what it means. As in statistical analysis, you have to tell the program which assumptions you want it to make. You do this by setting parameters. Some NEGOPY parameters may seem innocent enough, but they have serious implications for your analysis. This section of the manual explains these assumptions and the options available to you. It is your responsibility to understand your data and to make sure it fits the assumptions of the program. The chapter about relationships and measurement of data gives an introduction to the most important issues you will be faced with in this regard. In particular, you should be careful about the following areas:

- **Handling unreciprocated / unconfirmed links.**

When you are deciding whether to drop all one-way links or just the weak ones, or to keep them in the analysis, you are extending the operational definition of "relationship". It is possible that the modified operational definition will no longer provide a valid measure of the concept you were originally interested in. It is also likely that the results you obtain will vary, depending on which option you select here.

Some writers have objected to the fact that NEGOPY forces you to extend the operational definitions in this way. If you carefully examine any computer program for any kind of data analysis, you will see that NEGOPY is no different in this respect. The computer must do something with the data. You ought to both know what that "something" is and have control over how it works; if you don't, it is quite probable that the "something" will result in the implicit adoption of unwanted assumptions about your data.

It is common practice to run the program several times with different approaches to unreciprocated/unconfirmed links, and to use the most appealing results -- the ones that show a clean structure composed of several relatively small groups with few indirect liaisons. To do this is to choose the operational definition that produces the "nicest" results, regardless of the fact that it may conflict with the conceptual approach to the research situation. It would be better to choose operational definitions (which includes parameter settings specifying what counts as a link and what criteria must a set of nodes satisfy before you call them a group) that agree with your conceptual understanding of the situation you are studying. In any case, the link



In general, you should not change a parameter from its default value unless you know what the parameter does and why it is acceptable to change it. When you do alter some of the parameters mentioned in this section, you should state this in any articles you write using the analysis, and you should provide theoretical justification for the altered values.



The patterns NEGOPY looks for are based on a particular type of structural differentiation. When there is structural differentiation, some parts of the network will be structurally different from other parts -- they will have different structural characteristics, or they will play different roles in the way they relate to the rest of the network.

There are many ways in which parts of a network may be distinguished from one another. The types of distinctions NEGOPY makes are based on a set of definitions and criteria drawn from a hierarchical systems-theoretic view of organizations. The main conceptual basis for the category system employed by NEGOPY is that large complex systems are composed of subsystems ("groups"). A set of criteria which allow distinctions to be made is based on this starting point:

- a) Each subsystem must have the ability to function as a unit, which leads to the requirement that there must be an ability for the members of any subsystem to interact with the other members of the same unit.
- b) An individual who is more inside the group than outside the group, in terms of the interaction that individual has with other members of the network, is defined to be a member of the group<sup>7</sup>.
- c) If most of an individual's interaction is with members of groups but not with the members of any particular group, that individual is not a group member, but rather functions as a "linker" or "direct liaison".
- d) Individuals who have most of their interaction with other liaisons (not with members of groups) are classified as "indirect" or "multi-step" liaisons. They provide indirect connections between groups by connecting other liaisons.
- e) Individuals who have no contacts at all or who are minimally connected to others are categorized as "isolates". Those with no links at all are called "isolates Type 1". Those with only one link are "isolates Type 2" unless they both are linked only to each other, and then they are called "isolated dyads"<sup>8</sup>.

---

<sup>7</sup>Ordinarily the strength of links is considered for the classification of all non-isolate individuals. The program provides a fairly flexible ability to manage the strength of links by means of a variety of mathematical transformations of up to ten strength weights per link. If this ability is used to assign a strength of 1.00 to all links, all links will be considered to have the same strength. This means that only the number of links is considered for all role assignments. Note that this is an effective change of the role definitions.

<sup>8</sup>Note that the only consideration made when classifying an individual as an isolate is the number of

(continued...)



to describe a network and its members is probably the root of many of the problems people have with the program. The line between being a liaison and being a group member is even more imaginary than the walls around Les Nessman's office. The "link" connecting me to you isn't a physical bond. If there is a link, it is because of what we have perceived as a more-or-less stable on-going pattern of interaction that takes place over time. Notice how behavior has been transformed into structure. Think of the change from being someone's "friendly acquaintance" to being someone's "friend." Is there a discontinuity -- a point at which the situation changes, and you are no longer just an acquaintance? Or is it a gradual change that happens over a period of time? Finally, where does the change happen? What changes?

Whenever a discrete category "filter" is used to view a continuous reality, the resulting image is simpler, sharper, easier to work with, and easier to describe. It is also slightly less informative, because no distinctions can be made between the people who were close to the lines dividing categories and those who were in the middle or at the other end of their categories. If the discrete category system is mutually exclusive, people can only be in one category at a time. This means that overlapping categories are ruled out.

The discrete categorical system also causes problems because there may be more than one way of describing a network in terms of liaisons, groups, and isolates. What makes this particularly interesting is the fact that each of the descriptions, as different from the others as it may be, will completely satisfy the definitions and criteria used to define the network roles.

Shouldn't we expect problems when we try to make a picture of social reality when we first squeeze the dimension of time and variation over time right out of the analysis, and when we then chop up the what's left into discrete bits and pretend that what is left is no different from what we started with? What would these problems look like? The suggestion is that they won't look like what they are. Instead, they will crop up again and again, in a variety of places, in a range of manifestations.

NEGOPY's use of a discrete categorical system also causes problems because the descriptions of networks in terms of groups, liaisons, and isolates is a qualitative description, rather than a quantitative one. There is no doubt that this qualitative description is based on quantitative information about links, but links between people also involve qualitative information ("to **whom** do you turn for advice?"). The unusual mix of qualitative and quantitative information into a single analytic model makes it difficult to apply ordinary quantitative methods of examination and description. It also





cal, although various aspects of the topological structures are described in quantitative terms.

## **Finding Isolates**

Isolates and tree nodes are very easy to find. NEGOPY always finds all the isolates and tree nodes in any network, regardless of how the parameters that control group detection and testing are set. Because they are easy to find and because they have no effect on the placement of other nodes, the isolates are identified at the outset. They do not enter into the group detection or testing procedures, since they are already identified.<sup>10</sup>

## **Two stages of analysis**

NEGOPY uses a two-stage procedure to locate and identify groups, liaisons, etc. The first stage begins with an iterative process that constructs a global representation out of the pointillistic data that describes links between individuals, and ends with a procedure that identifies certain kinds of patterns in this global representation. The product of the first stage is a tentative description of the network, in terms of groups, liaisons, etc.

The second stage performs a number of logical and mathematical tests on the tentative description, revising the tentative description where appropriate.

For some networks, the first stage does most of the work, as the tentative description is not changed in the second stage. For most networks, both stages play an important role, as the tentative description is only an approximation to the final results, and must be modified by the second stage. In some cases, the first stage is unable to discern the patterns in the network, and the second stage does most of the work.

If the second stage is somehow inhibited or prevented from acting, the final results will depend heavily on the output of the first stage. If the first stage is inhibited or pre-

---

<sup>10</sup>Stohl and Kakarigi criticized the program because connections with isolates are not considered in the determination of whether or not a node is a group member. Their argument here is about the definitions the program begins with and the criteria it uses to distinguish between group members, liaisons, and isolates. This is like criticizing factor analysis that uses an orthogonal rotation because it forces the factors to be orthogonal to one another. It may be appropriate to use a different rotation (or a different set of definitions) for a particular research situation. If the definitions NEGOPY uses are not suitable for your research application, you should not use NEGOPY.



accurately describe their contacts, or to the use of a symmetrical measurement model with assymetric relationships. In any case, the program must be told how to deal with what may appear to be measurement error, and how to treat the two halves of confirmed or reciprocated links. These parameters should be set in accordance with the conceptual model you have of the relationships your data describes. While different settings for these parameters may result in a different network structure because they influence the number of links the program has to deal with, they do not alter the way the group detection procedures behave.

One parameter (**P10, multiplelinks**) tells the program how it should deal with situations in which one person reports more than one link with a second person. Is this an error in data entry, or does the method of recoding links make this a normal occurrence?

This chapter does not address these parameters. An assessment of what happens when you vary the parameter that specifies the minimum link strength may be interesting, but the tremendous variability in networks and in the ways researchers measure and describe networks in terms of data makes such an analysis useless.

An assessment of what happens when you decide to keep or delete one-way links is also out of place, but for a different reason: what you decide to do with this issue should be a function of what you define a "link" to be, which should follow from how you conceptualize the relationship you are interested in studying. If you do two runs on your data, one keeping one-way links and one discarding them, you will likely see a large difference in the results. If you then choose the results you like best and work backwards from there to determine how you should conceptualize and operationalize the relationship your network is based on, you are "cheating". This is like doing a few dozen t-tests and then selecting hypotheses that relate only the variables you already know to be statistically related to one another.

## Parameters that control the output

Twelve parameters control the amount, type, and organization of information the program produces as output<sup>12</sup>. They have no effect whatsoever on the results of the analysis. They are therefore not included in the tests described here.

---

<sup>12</sup> **P13** -- 132-columnprint; **P14** -- nrawprint; **P15** -- linkinformation; **P16** -- isolate-information; **P17** -- densityhistogram; **P18** -- testinformation; **P19** -- groupinformation; **P20** -- auxfileoutput; **P21** -- monitorinformation; **P22** -- dmatrixprint; **P23** -- onefileoutput.



## The important one

There is one final parameter, **P28 %withingroup**, which plays a critical role in the operational definition of groups, group membership, and liaisons. Since this parameter does not affect the number of times the program will do something, or the size something must be before it will be tested, or the severity of the test, it should not be seen as controlling the operations the program performs. However, because this parameter specifies the point beyond which "inside the group" becomes "outside the group", it is central to how the program defines a group.

If the value you supply for **P28** is not consistent with your definition of what a group is, the results you get will be invalid, in the sense that what the program calls a "group" will not correspond to what you call a "group", although you won't be able to tell this from the output the program produces.

You should expect the program to produce different results if you change the value of **P28**, because changing this parameter means you are changing your definition of what a group is. It is not possible to predict what effect these changes would have for your data, as each network is unique. The main constraint on the way you set **P28** is that it must be greater than 50% because NEGOPY does not have the ability to handle overlapping groups.

## Pragmatic issues

*how the program behaves in different circumstances, how robust it is,  
how sensitive it is, etc*

Three sets of data were used for the bulk of the testing whose results are described here:

- a) "SK" -- the eight-person binary network from the "Critical Appraisal" paper (Stohl & Kakarigi, 1985);
- b) "EQN" -- the 33-person "DATA.EQN" that is supplied with the NEGOPY program;
- c) "249" -- data from a study of a pulp and paperworkers' union with 249 members. This data is also supplied with the NEGOPY program to demonstrate the program and to verify that the program is working correctly.

The SK data is by far the simplest. It was created by Stohl & Kakarigi to test and demonstrate the program's performance. A sociogram for this network is shown in



Kakarigi is to be commended for the work he did. Unfortunately, there were a small number of bugs in the translation. One of these systematically deleted certain links from the network. Had the authors run the program on the test data that was supplied, they would have been surprised to see the program stop dead in its tracks long before completing the analysis.

Because the SK data is small and simple and relatively well-known, it is still a good starting point for an evaluation of the program's performance. The evaluation included the seven parameters that influence the detection and testing of groups. **P28, %within-group**, the parameter that sets the critical percentage used in the definition of groups, was not included in the test. To recognize the small size of the SK network, **P29** was set to a value of 5, instructing the program to perform all tests on groups having five or more members.

### S&K Results

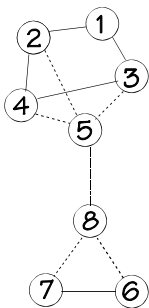
0. The program was run in a "default" mode with only the data description and output control parameters are set and the parameters controlling the detection, definition, and testing of groups are left at their default values. The "default" solution for this network is "correct" in the sense that it is the solution one would want the program to produce if it were working correctly. It has two groups, one with three members and one with five members. No liaisons or isolates of any kind.
1. **P24**, number of iterations, had no effect whatsoever, as long as it was not below 1. The program was run with all values from 1 to 100. The default value for this parameter is 4. When this was the only parameter altered, the program always produced the default network.
2. **P25, scanradius**, the width of the scanning window used in initial group detection, had no effect between values of 10 and 1000. As long as the program found at least one group in the first stage of analysis, the final results were correct -- they matched the default 2-group solution. The reason the program needed to find at least one group was that if no groups were found, the program would skip to the final routines that print the results. To account for the possibility of there being groups when the initial analysis found none, I modified the program so it always tries to form groups out of the nodes not included in groups found by the initial process. After this modification, it didn't matter how P24 was set for this data.
3. **P26, sensitivity**, the sensitivity of the initial group detection process, was tested





Since removing 2 members would have a greater chance of splitting the group apart than removing one, you would expect the program to find more smaller groups if you set **P31** to a value like 3, 4, or 5. However, this isn't how it works.

The SK data illustrates how this is the case. Figure 2 shows their network with the two most central members, nodes #5 and #8, temporarily removed.



**Fig. 2** S&K network minus two most central nodes

What was originally one group now splits into two fragments, the first including nodes #1, #2, #3, and #4, and the second one including nodes #6 and #7. **NEGOPY** looks at the size of the second-largest fragment (in this case, 2) and compares it with the number of nodes removed for the test (also 2). If the second-largest fragment isn't at least twice as large as the number removed, the program decides that the group is actually one group, and it puts it back together. This is done because in many cases the removal of the most central members would cause one large fragment and a few very small fragments to form. The program would then test each of the fragments again, often "chipping" another small fragment off of the main group. In some cases this reduced fairly large groups to a large number of tiny fragments. The test that compares the size of the second-largest fragment with the number of nodes removed prevents this from happening, while it effectively breaks apart groups that are actually multiple groups stuck together with a few bridge links.

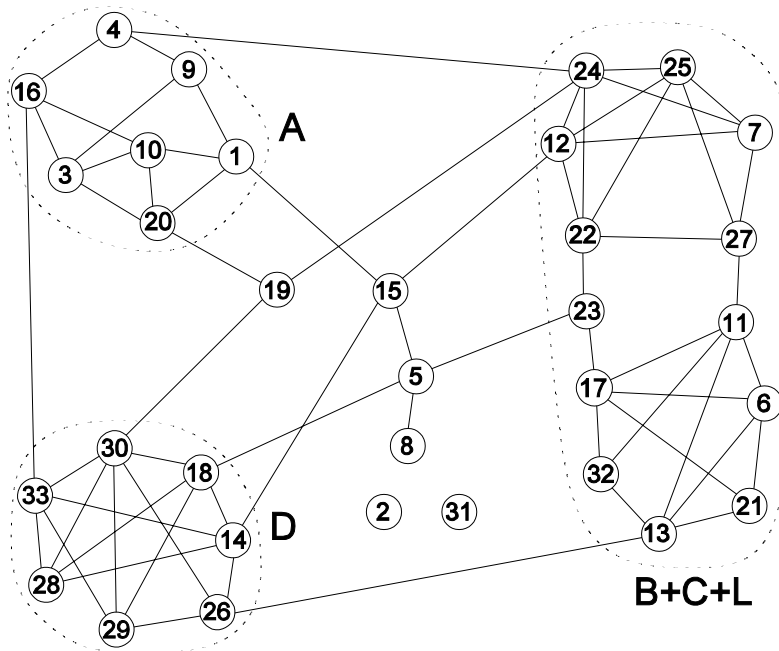
With the SK data, all values of **P31** greater than 1 resulted in the detection of one single group that includes all eight nodes. This is because values greater than 1 effectively disabled the group testing for this small network.

### S&K Summary

The SK data was not an adequate test of the program because it did not test the first stage of analysis and it was not large enough to allow the full range of variations found in most networks to come into play. The fact that the parameters which control the first stage of analysis were virtually irrelevant for this data points to the robust nature of the program's second stage of analysis when faced with simple, "clean" networks with sharp structural differentiation. Unfortunately, few real networks are like this.



liaisons) when there were between 4 and 53 iterations. It also gave the same results with 2 iterations. When there were 3 iterations, or when there were between 54 and 200 iterations, a different result was obtained. This one had three groups and three



**Fig. 4** The “A, B+C+L, D” three-group solution

liaisons.

For purposes of description, the four groups in the correct run were named “A”, “B”, “C”, and “D”. The second of the groups in the 3-group solution was a combination of groups B and C (the two 6-person groups) plus one of the liaisons (“B+C+L”). The first and third groups were the same as groups A and D. Figure 4 shows the 3-group solution (A, B+C+L, D).

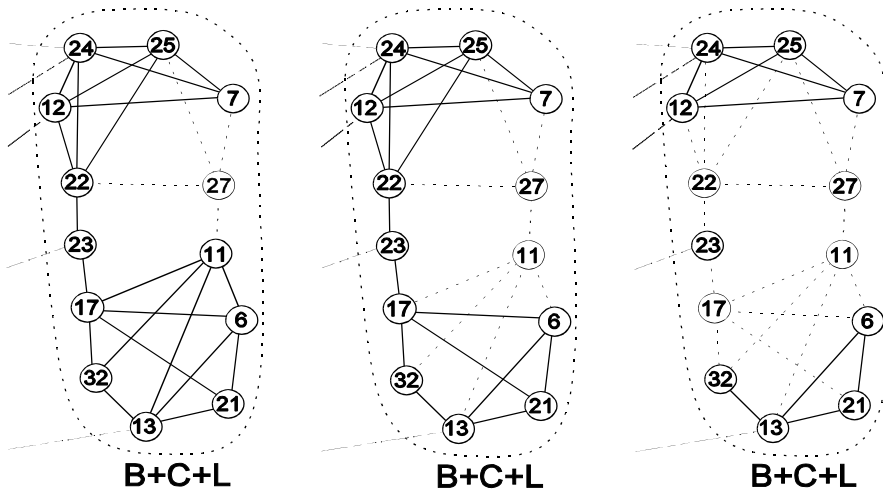
If **P25**, the scan radius parameter, is reduced from its default of 200 to 100 while the number of iterations was set to give the 3-group solution described above, the program gave the original 4-group solution instead. The scan radius parameter thus works together with the number of iterations.

2. **P25**, the scan radius parameter, was varied from 10 to 700 (the default is 200).



D) solution, the default settings for the parameters that control group testing are not sufficient for the program to split the “B+C+L” group apart into B, C, and L. Additional runs were done to see what effect **P31** would have on this situation. Would it be possible to “force” the program to split the “B+C+L group apart?”

With **P31** set to 1, 2, 3, or 4, the program always produced the 3-group solution. Why this happens is shown in Figures 6, 7, & 8. The program removes nodes in order of centrality. The one with the shortest average distance to others in the group is removed first, etc. For this network, the most central node is #27. Following are #11, #22, and #17. Removing #27 fails to split the group into disconnected segments. Removing #27 and #11 also fails to split the group apart. When #22 and #17 are also removed, the rest of the group splits into a number of small disconnected fragments. Because these fragments are so small compared to the number of nodes removed, the program determines that it is a “bad split” and the group is put back together.



**Figs. 6, 7, 8.** The effect of setting P31 to higher values with the EQN 3-group solution

The problem is that removing #27, #11, and #22 (or #27, #11, #22, and #17) leaves connected clusters smaller than the part removed. In larger networks where the groups tend to be larger, changes to P31 like the ones described here may prove to be more fruitful. In order to have a “good” split when 4 nodes have been removed, the group must have an original size big enough to leave two fragments with at least 8 members each. In other words, it must have at least  $8 + 8 + 4 = 20$  members



1. **P24**, number of iterations, was varied from 2 to 20. The table below summarizes the results.
2. Fewer than 4 iterations wasn't enough for the program to construct a global representation of the network that allowed the groups to be identified by the first stage of analysis. When large numbers of iterations were done, the accumulation of rounding error led to a loss of resolution and the collapsing of several smaller groups into a small number of large groups. These small groups were connected by between-group links and links to a relatively large number of nodes classified as liaisons in the default run.

By setting **P25, scanradius**, to relatively low values (50 to 100), the resolution at high numbers of iterations was greatly improved. This made it possible for the program to find a larger number of smaller groups in the initial detection phase, which increased the likelihood of resolving groups out of the relatively undifferentiated areas of the network.

parameter settings <sup>13</sup>	# groups	# same <sup>14</sup>	# grp membs	# liaisons
default	14	--	81	36
2 iterations	12	10	93	24
3 iterations	11	10	91	26
5 iterations	13		58	59
8 iterations	14	9	72	45
10 iterations	9	6	111	6
20 iterations	6	5	112	5
20 iters, scan=73	14	7 (2,2) <sup>15</sup>		
3 iters, scan=150	11	10 (2)	101	16
4 iters, scan=15	14	11 (2+)	91	26

<sup>13</sup>All group detection and testing parameters other than the one mentioned remain set to their default values.

<sup>14</sup>"# same" refers to the number of groups in the results of this run that were identical with groups produced in the default run.

<sup>15</sup>The "(2,2)" means that two of the groups in this run were each composed of two groups in the default run. In other words, there were four groups in the default run represented as two groups in this run.





other words, such a test reflects more on the particular set of data than on the program itself.

## **Discussion**

The weakest step in the use of NEGOPY for network analysis is the bridge between the conceptualized relationship under investigation and the data that is collected and submitted to the program for analysis. Human relationships are among the most complex phenomena that we can attempt to investigate. They are multidimensional and non-linear in complex ways. They can be approached from several levels of abstraction, and more sophisticated views of social reality indicate that internal inconsistencies in the data should be expected and should not be treated as errors. The result is that over-simplified approaches to conceptualization and measurement continue to be used, resulting in weak data that should not be expected to produce robust, valid, or reliable results. However, this is not a problem with NEGOPY, but rather a problem due to a lack of understanding of network data and measurement.

NEGOPY has been criticized as "ad hoc," even though it is based on a clear set of definitions and criteria drawn from a systems theoretic view of complex organization, it uses these definitions through all stages of analysis and classification of network members, and it delivers final results expressed in terms of these conceptual definitions. This is not ad hockery. When researchers systematically chip away at their data until they get a picture they like, there is a problem. Note that the problem is due to a combination of the nature of the data and the way the program is used, rather than with the program itself.

To guide against this kind of problem, users should have a clear understanding of what counts as a link. Does it have to be reciprocated or confirmed? How strong does it have to be before it counts as a link? This must be considered in the light of what the program is looking for -- groups of individuals who have most of their contact with one another. When the link is understood as a component in this larger context, does its nature change? Once these questions have been answered, the user should set the program's parameters to agree with what they have decided their data represents and with what counts as a link. Taking this approach will reduce the need to work backwards from parameter settings to the conceptual definitions that are consistent with what was done with the data.

Network analysis is more complex than it has seemed. Much of the complexity comes in in the process of operationalization and measurement. Many of the assumptions



## References

- Barnett, G. (1973). "On the nature of random structure". Department of Communication, Michigan State University, East Lansing.
- Kincaid, D.L. (1993). "Communication Network Dynamics: Cohesion, Centrality, and Cultural Evolution." In W. Richards, and G. Barnett (Eds.) *Progress in Communication Sciences XII*. (pp. 111-134). Norwood: Ablex .
- Richards, W.D. (1988). "The NEGOPY Network Analysis Program" Connections, the Bulletin of the International Network for Social Network Analysis
- Richards, W.D. (1986). "The NEGOPY Network Analysis Program User's Manual" School of Communication, Simon Fraser University, Burnaby, BC CANADA
- Rogers, E. M., & Kincaid, D. L. (1981). *Communication Networks: Toward a New Paradigm for Research*. New York: Free Press.
- Seary, A.J. & Richards, W.D. (1995). "Partitioning Networks by Eigenvectors." Presented to European Network Conference, London, July 6-10. Published in conference proceedings. Available at <http://www.sfu.ca/~richards>
- Stohl, C. & Kakarigi, D. (1985). "The NEGOPY Network Analysis Program: A Critical Appraisal" paper presented to SCA and circulated in a revised version

## Related readings

- Alba, R.D. (1972), "COMPLT: A Program for Analyzing Sociometric Data and Clustering Similarity Matrices", *Behavioral Science*, 17, 566–567.
- Alba, R.D. (1973), "A Graph-Theoretic Definition of a Sociometric Clique", *J. Math. Soc.*, 3, 113–126.
- Alba, R.D. and M.P. Guttman (1974), *SOCK: A Sociometric Analysis System*. New York: Columbia University Report.
- Alba, R.D. and G. Moore (1983), "Elite Social Circles", in R.S. Burt and M.J. Minor (eds.), *Applied Network Analysis*. Beverly Hills, CA: Sage.
- Allen, T.J. and S.I. Cohens (1969), "Information Flow in Research and Development Laboratories", *Administrative Science Quarterly*, 14, 12–20.
- Alt, J.E. and N. Schofield (1975), "CLIQUE: A Suite of Programs for Extracting Cliques from a Symmetric Graph", *Behavioral Science*, 20, 134–135.
- Arabie, P. (1977), "Clustering Representations of Group Overlap", *J. Math. Soc.*, 5, 113–128.
- Arabie, P., S.A. Boorman and P.R. Levitt (1978), "Constructing Blockmodels: How and Why", *J. Math. Psych.*, 17, 21–63.



- tal Uncertainty and Communication Network Structuring", International Communication Association, New Orleans.
- Danowski, J.A. (1984), "Personal Network Integration: Infographic, Psychographic and Demographic Characteristics", paper presented to information/systems division, International Communication Association, San Francisco, May.
- Davis, J.A. (1967), "Clustering and Structural Balance in Graphs", *Human Relations*, 20, 181–187.
- Davis, J.A. (1977), "Sociometric Triads as Multi-Variate Systems", *J. Math. Soc.*, 5, 41–60.
- Davis, J.A. and S. Leinhardt (1972), "The Structure of Positive Interpersonal Relations in Small Groups", in J. Berger, M.J. Zelditch and B. Anderson (eds.), *Sociological Theories in Progress*, Vol. 2. New York: Houghton-Mifflin.
- Delattre, M. and P. Hansen (1980), "Bicriterion Cluster Analysis", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-2, 277–291.
- Devereux, J., P. Haeberli and O. Smithies (1984), "A comprehensive set of sequence analysis programs for VAX", *Nucleic Acids Res.*, 12:1, 378–395.
- Dill, W.R. (1958), "Environment as an Influence on Managerial Autonomy", *Administrative Science Quarterly*, 2, 409–443.
- Doreian, P. (1980), "Linear Models with Spatially Distributed Data: Spatial Disturbances or Spatial Effects", *Soc. Methods & Res.*, 9, 29–66.
- Doreian, P. (1982), "Maximum Likelihood Methods for Linear Models: Spatial Effect and Spatial Disturbance Terms", *Soc. Methods & Res.*, 10, 243–270.
- Dubin, R. and S. Spray (1964), "Executive Behavior and Interaction", *Industrial Relations*, 3, 99–108.
- Edwards, J.A. and P.R. Monge (1977), "The Validation of Mathematical Indices of Communication Structure", *Communication Yearbook*, 1, 183–194.
- Eisenberg, E.M., R.V. Farace, P.R. Monge, E.P. Bettinghuis, R. Kurchner-Hawkins, K. Miller and L. Rottman (1985), "Communication Linkages in Inter-organizational Systems: Review and Synthesis", *Progress in Communications Sciences*, 6, 236–261.
- Farace, R.V. and D. Johnson (1974), "A Comparison of Selected Network Characteristics Across Six Organizations", paper presented to annual meetings of International Communication Association, New Orleans.
- Farace, R.V. and T. Mabee (1980), "Communication Network Analysis Methods", in P.R. Monge and J.N. Cappella (eds.), *Multivariate Techniques for Human Communication Research*. New York: Academic Press, pp. 365–392.
- Farace, R.V., P.R. Monge and H.M. Russell (1977), *Communicating and Organizing*. Menlo Park, CA: Addison-Wesley.
- Farace, R.V. and M. Pacanowsky (1974), "Organizational Communication Role, Hierarchical Level and Relative Status", paper presented to the Academy of Management Association, Seattle.
- Felling, A.J.A. (1975), "A Graph-Theoretical Approach to the Structure of Local Elites", *Zeitschrift für Soziologie*, 4, 221–233.
- Felling, A.J.A. and T. van der Weegen (1976), "Programmer's Notes for Main Program NCLIQUE". Nijmegen, Netherlands: Mathematical Soc. and Research Technical Department, University of Nijmegen.
- Fienberg, S.E. (1979), *The Analysis of Cross-Classified Categorical Data*. Cambridge, MA: MIT Press.



- Kochen, M. (1985), "The Structure of Acquaintance Nets and Rates of Societal Development", *Social Networks*, 7:4, 323–339.
- Krippendorff, K. (1980), "Clustering", in P.R. Monge and J.N. Cappella (eds.), *Multivariate Techniques for Human Communication Research*. New York: Academic Press, pp. 259–308.
- Kroger, M. and A. Kroger-Block (1984), "Simplified computer programs for search of homology within nucleotide sequences", *Nucleic Acids Res.*, 12:1, 193–213.
- Kruskal, J.B. and M. Wish (1978), *Multidimensional Scaling*. Beverly Hills, CA: Sage.
- Lankford, P.M. (1974), "Comparative Analysis of Clique Identification Methods", *Sociometry*, 37, 287–305.
- Laumann, E.O. (1971), *Bonds of Pluralism*. New York: Wiley Interscience.
- Lesniak, R., M.P. Yates, G.M. Goldhaber and W.D. Richards, Jr. (1977), "NETPLOT: An Original Computer Program for Interpreting NEGOPY", paper presented to the International Communication Association, Berlin.
- Levine, S., P.E. White and B.D. Paul (1963), "Community Interorganizational Problems in Providing Medical Care and Social Service", *Am. J. Public Health*, 53, 1183–1195.
- Lingoes, J. (1973), "The Guttman-Lingoes Nonmetric Program Series". Ann Arbor, MI: Ma thesis Press.
- MacRae, Jr., D. (1960), "Direct Factor Analysis of Sociometric Data", *Sociometry*, 23, 360–371.
- McFarland, D.D. and D.J. Brown (1973), "Social Distance as Metric: A Systematic Introduction to Smallest Space Analysis", in E.O. Laumann (ed.), *Bonds of Pluralism: The Form and Substance of Urban Social Networks*. New York: Wiley, pp. 213–253.
- McQuitty, L.L. (1957), "Elementary Linkage Analysis for Isolating Orthogonal and Oblique Types and Typal Relevancies", *Ed. & Psych. Measurement*, 17, 207–229.
- Meyer, M.W. (1972), "Size and the Structure of Organizations: A Causal Analysis", *Am. Soc. Review*, 37, 434–441.
- Miles, R. (1980), *Macro Organizational Behavior*. Santa Monica, CA: Goodyear.
- Monge, P.R., K.K. Kirste and J.A. Edwards (1974), "A Causal Model of the Formation of Communication Structure in Large Organizations", paper presented at International Communication Association, New Orleans, May.
- Mullins, N.C. (1968), "The Distribution of Social and Cultural Properties in Informal Communication among Biological Scientists", *Am. Soc. Review*, 33, 786–797.
- Murray, Stephen O. and R.C. Poolman (1982), "Strong Ties and Scientific Literature", *Social Networks*, 4:3, 225–232.
- Ornstein, M.D. (1982), "Interlocking Directorates in Canada", *Social Networks*, 4, 3–25.
- Pattee, H.H. (1973), *Hierarchy Theory*. New York: Braziller.
- Pfeffer, J. and G.E. Salancik (1978), *The External Control of Organizations: A Resource Dependence Perspective*. New York: Harper & Row.
- Phillips, D.P. and R.H. Conviser (1972), "Measuring the Structure and Boundary Properties of Graphs: Some Uses of Information Theory", *Sociometry*, 35, 235–254.
- Porter, L.W. and K.H. Roberts (1973), "Communication in Organizations", Tech. Rep. No. 12. Arlington, VA: ONR, ERIC (microfiche) ED-066-773.
- Pugh, D.S., D.F. Hickson, C.R. Hinnings and C. Turner (1968), "Dimensions of Organizational Structure", *Admin. Science Quarterly*, 13, 65–104.
- Reid, W. (1967), "Interorganizational Co-ordination in Social Welfare: A Theoretical Approach to





- Rogers, E. (1962), *The Diffusion of Innovations*. Glencoe, IL: Free Press.
- Rogers, E.M. and D.L. Kincaid (1981), *Communication Networks: Toward a New Paradigm for Research*. New York: Free Press.
- Rosengren, W.R. (1967), "Structure, Policy, and Style: Strategies of Organizational Control", *Administrative Science Quarterly*, 12, 140–164.
- Russell, H. (1974), "Communication Network Study", memorandum to Department of Agriculture, Victoria, Australia, 11 October.
- Sailer, L.D. (1978), "Structural Equivalence: Meaning and Definition, Computation and Application", *Social Networks*, 4, 117–145.
- Schwartz, D.F. (1969), "Liaison Roles in the Communication Structure of a Formal Organization: A Pilot Study", paper presented at annual meeting of National Society for the Study of Communication, Cleveland.
- Schwartz, J.E. (1976), "An Examination of CONCOR and Related Methods for Blocking Sociometric Data", in D.R. Heise (ed.), *Sociological Methodology*. San Francisco, CA: Jossey- Bass, pp. 255–282.
- Seidman, S., and B. Foster (1979), "SONET-1", *Social Networks*, 2, 85–90.
- Seidman, S.B. and B.L. Foster (1978), "A Graph-Theoretic Generalization of the Clique Concept", *J. Math. Soc.*, 6, 139–154.
- Shapiro, B.A., J. Maizel, L.E. Lipkin, K. Currey and C. Whitney (1984), "Generating non-overlapping displays of nucleic acid secondary structures", *Nucleic Acids Res.*, 12:1, 75–100.
- Shaw, Jr., W.W. (1983), "Statistical Disorder and the Analysis of a Communication Graph", *J. Am. Society for Information Science*, 34, 146–149.
- Shepard, R.N. (1972), *Multidimensional Scaling*. New York: Seminar Press.
- Sondquist, J. and T. Koenig (1975), "Interlocking Directorates in the Top U.S. Corporations: A Graph Theory Approach", *Insurgent Sociologist*, V:III, 196–229.
- Stork, D., and W.D. Richards. (1992). "Nonrespondents in Communication Network Studies: Problems and Possibilities", *Group and Organization Management*, June, 17:6, 193-211.
- Thomason, G.F. (1966), "Managerial Work Roles and Relationships: Part I", *J. Mgmt. Studies*, 3, 270–284.
- Thompson, J.D. (1956), "Authority and Power in 'Identical' Organizations", *Am. J. Soc.*, 62, 290–301.
- Tryon, R.C. and D.E. Bailey (1970), *Cluster Analysis*. New York: McGraw-Hill.
- Warner, W.L. D.B. Unwalla, and J.H. Trimm (eds.) (1967), *The Emergent American Society: Large-Scale Organizations*. New Haven: Yale Univ. Press, pp. 121-157.
- Wasserman, S. (1980), "Analyzing Social Networks as Stochastic Processes", *J. Am. Statistical Association*, 75, 280–294.
- Watanabe, K., K. Yasukawa and K. Iso (1984), "Graphic display of nucleic acid structure by a micro-computer", *Nucleic Acids Res.*, 12:1, 801–856.
- Webber, R.A. (1970), "Perceptions of Interactions between Superiors and Subordinates", *Human Relations*, 23, 235–248.
- Weick, K. (1969), *The Social Psychiatry of Organizing*. Reading, MA: Addison-Wesley.
- Weineke, P.R., E. Mansfield, D.A. Jaffe and D.L. Brutlag (1984), "Rapid searches for complex patterns in biological models", *Nucleic Acids Res.*, 12:1, 263–280.
- Weinberg, A., L. Ullian, W.D. Richards and P. Cooper. (1981). "Informal Advice and Information-seeking Between Physicians", *Journal of Medical Education*, March , pp. 174–180.



**Index**

- 132-column print
  - parameter ..... 74
- 2-Step Weight
  - parameter for controlling ..... 81
- A**
- ad hockery ..... 161
- Assigning ID numbers ..... 119
- Assumptions
  - ratio-level scaling ..... 43
- Auxiliary File output
  - parameter for controlling ..... 76
- Auxiliary Output file ..... 62
- Auxilliary output file
  - contents and organization ..... 62
  - dump of internal data structure .... 67
  - summary ..... 66
- B**
- Bernard, Russ ..... 31
- Big groups ..... 130
- BKS ..... 31
- Blockmodelling compared to NEGOPY 14
- C**
- Cautions ..... 136
  - changing operational definitions .. 136
  - impact of deleting weak links .... 137
  - unreciprocated/unconfirmed links 136
- Changing the role definitions ..... 28, 77
- Cliques ..... 8
  - Groups ..... 8
- conceptual basis for category system used
  - by NEGOPY ..... 141
- Confirmation ..... 20, 33
  - Reciprocation ..... 20
- Confusion and counting links .... 41, 90
- Connectiveness testing ..... 26
- content/function and group detection 131
- Continuous measures ..... 14
- Controlling Group Detection and Testing
  - parameter settings ..... 126
- Controlling the Output ..... 125
  - parameter settings ..... 125
- Counting links ..... 37, 88
- D**
- Data
  - accuracy ..... 31
  - behaviour ..... 32
  - context ..... 32
  - format of data file ..... 57
  - logic of descriptions ..... 31
  - matrix ..... 34
  - multiple strength indicators ..... 43
  - relationships ..... 32
  - symmetry ..... 33
  - validity ..... 30, 32
  - verification of input ..... 82
- Default Parameter Definition file .... 54
  - SeeAlso Input/Output ..... 54
- Default values for parameters ..... 8, 25
- Definitions
  - Changing the Role Definitions ... 28
  - definition of "group" and
    - "liaison"parameter ..... 79
  - definition of group ..... 141
  - definition of network positions . 141, 142
  - distance matrix ..... 27
  - group member ..... 10
- Density Histogram ..... 24
  - control of printing ..... 75
  - Group Detection ..... 24
  - use and interpretation ..... 126
- Direct Liaison
  - SeeAlso Liaison Type 1 ..... 10



distance matrix	27	parameters	71
group splitting process	27	ID Numbers	
maximum group size	28	assigning ID numbers	17
membership	12, 26	group detection process	22
P24 - P31	28	range	19
standard deviation	28	imprecision of NEGOPY	140
sub-groups	27	inconsistency of NEGOPY	140
Groups		information about groups	
connected	9	controlling amount of	76
definition	9	information about isolates	
Group Detection Process	22	controlling amount of	75
logical tests	17	information about links	
maximum size	28	controlling amount of	75
members	10	information about program's execution	
one big group	130	parameter for controlling	76
output descriptions of Groups	104	information about tests	
overlapping	12	controlling amount of	75
when none are found	13	Input/Output	52
<b>H</b>		role definition documentation	54
Header lines	59	Unit 1 -- default parameter definition	
Hierarchical clustering	15	file	52
Highest ID Number	71	Unit 9 -- Data file	57
How negopy categorizes members of		Input/Output files	
networks	9	Auxiliary Output file	62
How NEGOPY categorizes members of		Default Parameter Definition file	54
networks		Namelist file	55
SeeAlso Role Definitions	9	Parameter Setting file	59
How to count links	37, 88	Standard Output File	61
How to interpret output	82	Interconnectiveness & structural	
How to tell NEGOPY which files you		differentiation	101
want it to use		Isolates	
.....	53	defined	9
How to use NEGOPY	118	Isolate Type 1	9
How to verify data input procedures	88	Isolate Type 2	9
<b>I</b>		Tree Node	10
ID Number		Isolates and tree nodes	
in data	13	identification of	145
		Iterative process	17, 22



**O**

One big group .....	130
one-way links	
deleting .....	147, 160
parameter .....	72
operational definitions	
parameter settings .....	136
parameters that control logical testing	
of groups	
.....	139
weak links .....	137
Output	
Analysis of Distance Matrix .....	105
bad strength equation .....	92
Density Histogram .....	102
discrepancy scores .....	96, 97
Distance Matrix .....	104
Groups as Units .....	104
Information about Group Detection	101
link description -- maximum detail .	97
link description -- medium detail .	95
listing of parameter settings .....	84
Measure of Structural Differentiation .	101
Processing ID numbers .....	91
Processing Multi-stranded links ...	92
Processing unconfirmed links .....	94
Raw Data Listing .....	87
Reciprocation analysis table .....	98
rejected data .....	91
Role definition summary .....	86
Strength Distribution Histogram ...	99
tentative group structure listing .	104
The links of individuals .....	95
Output File	
splitting the .....	76
Output information	
auxilliary output file contents .....	62

link strength discrepancies .....	21
Standard output file .....	61
Summary Tables .....	29

**P**

P29, minimumsplitsize	
prevents complete analysis .....	150
Parameter Setting file .....	59
header lines .....	59
Parameters .....	70
and operational definitions .....	136
Changing the Role Definitions ...	28
confirmation/reciprocation rates .	146
default specification file .....	19
Default values .....	25
effect of changing P11 Weakestlink...	130
effect of changing P25 Scanradius .	127,128
effect of changing P26 Sensitivity..	29,130
Lowest ID number .....	120
P01 EQN Number of nodes .....	120
P01 highestidnumber .....	146
P01 Number of nodes .....	120
P02 EQN Lowest ID number .....	120
P02 lowestidnumber .....	146
P04 EQN Name width .....	124
P04 Name width .....	124
P05 EQN Reciprocation ...	122, 125
P05 Reciprocation .....	20, 122, 146
P06 EQN Unrecipmin .....	122, 125
P06 Unrecipmin .....	20, 122
P06 unreciprateminimum .....	146
P07 EQN Mean strength set .....	125
P07 Mean strength set .....	125
P07 meanstrengthset .....	146
P08 Direction .....	121
P08 EQN Direction .....	121
P10 EQN Multiple links .....	125
P10 Multiple Link Option ...	19, 57





tion, testing of groups . . . . .	148	Reality . . . . .	30
Parameters that control output . . . . .	74	Reciprocation . . . . .	20, 33
132 or 80 column print . . . . .	74	Confirmation . . . . .	20
amount of information about groups	76	parameter . . . . .	72
amount of information about isolates	75	Rejected data	
amount of information about links . .	75	bad contact ID # . . . . .	91
amount of information about program		bad respondent ID # . . . . .	91
e x e c u t i o n . . . . .	76	bad strength equation . . . . .	92
amount of information about tests . .	75	rejecting weak or unconfirmed links . .	130
Auxiliary file . . . . .	76	Relationships	
density histograms . . . . .	75	asymmetric . . . . .	33
distance matrices . . . . .	76	communication . . . . .	20
number of lines of data to print . . .	74	content or function . . . . .	35
split the output file . . . . .	76	directed & undirected . . . . .	20, 33
Parameters that describe your data . . .	146	links vs relationships . . . . .	30
Participants		pairwise . . . . .	32
defined . . . . .	10	representations of relationships . . .	31
Group Members . . . . .	10	see also Strength . . . . .	36
Liaison Type 1 . . . . .	10	transitivity . . . . .	34
Liaison Type 2 . . . . .	10	types of relationships . . . . .	30
pattern-recognition methods . . . . .	144	Role Definitions	
Pattern-recognition process . . . . .	17, 22, 23	changing the definitions with P28 . .	12
Group detection . . . . .	22	See also Definitions . . . . .	9
Performance and Parameter Sensitivity	140	Running NEGOPY . . . . .	135
Conceptual Issues . . . . .	140	filename may not exceed 24 characters	134
iterative process . . . . .	145	how you associate files with the	
Operational: issues . . . . .	144	program . . . . .	135
Pragmatic issues . . . . .	149	<b>S</b>	
Pre-processing of data . . . . .	19	Sailor, Lee . . . . .	31
Preparation of Data File . . . . .	119	Scan Radius	
Preparing the Parameter Specification File	120	parameter . . . . .	78
problems		Scanning window . . . . .	24, 25
one big group . . . . .	130	Group Detection . . . . .	24
<b>R</b>		Parameter P25 . . . . .	24
Random samples and NEGOPY . . . . .	13	second stage of analysis	
Ratio-level data		logical and mathematical tests . . .	145
Strength . . . . .	13	sensitivity of the group detection process	

parameter	78	testing link strength	91
Setting Parameters		unconfirmed links	20
directed or undirected analysis	121	strength of links	
highest ID number	120	highest and lowest	125
ID numbers	120	managing	124
lowest ID number	120	multiple links	125
Confirmed & Unconfirmed links	121	parameter settings	124
controlling the output	125	Strength Transformation Equation	
Directed or Undirected data	120	as a parameter	81
Forcing symmetry	122	Strength Transformations	
Group Detection	126	combining indicators	43
link strength & confirmation	122	combining multiple weights	46, 48
link strength indicators	124	data	44
means vs. sums	122	elapsed-time scales	47
multi-stranded Links	123	exponentiation	45
namelist	124	scale reversal	45
number of Links	124	setting up strength transformation	
to control the density histogram	127-129	e q u a t i o n s	48
unreciprocated or unconfirmed links	122	Why do strength transformations?	45
weak one-way links	123	Structural Differentiation	101, 141
small networks	79	Symmetricality	33
parameter for controlling	79	relationships	33
split apart group		Symmetry	
parameter for controlling	80	and data	33
split apart group, amount to remove		and relationships	33
parameter for controlling	80	<b>T</b>	
Standard Deviation		tentative description of the network	
Group Testing Procedures	28	iterative process	145
Standard output file	61	logical and mathematical tests	145
Strength	13	transitive links and group detection	131
discrepancies	21	Transitivity	34, 102
group detection	22	relationships	34
group membership	12, 26	Transitivity Weight	
links	36	parameter for controlling	80
multiple indicators	46, 57	Tree node or structure	10
operationalizations	36	two-Step Weight	
ratio-level scaling	13, 43	parameter for controlling	81
strength weights	57		

**U**

Unconfirmed links .....	122
Unrecip Minimum	
parameter .....	72
unreciprocated (or unconfirmed) links	
parameter .....	72
unreciprocated (unconfirmed) links	
parameter .....	72
Unreciprocated links .....	122
unreciprocated or unconfirmed links .	122
unreliability of NEGOPY .....	140
Using the Distance Matrix .....	133

**V**

## Versions of NEGOPY

IBM PC .....	16
EQN, 1985 - ?? .....	15
Very Small Networks .....	132
P29, minimumsplitsize .....	150
parameter settings .....	132

**W**

WARNING .....	77
Changing P28 .....	28
weak links	
deleting .....	137
see also Strength .....	46
weak one-way links .....	123
What is analyzed in NEGOPY? .....	30
Within-Group links, mininum	
parameter .....	78

**Z**

zone overlap	
Group Detection .....	23