MDS Analysis Using Permap 11.8

Dr. Ronald B. Heady, University of Louisiana at Lafayette (retired)
Dr. Jennifer L. Lucas, Agnes Scott College
January 24, 2010

TABLE OF CONTENTS

1.0 OVERVIEW

1.1 Permap

The fundamental purpose of Permap is to uncover hidden structure that might be residing in a complex data set. Compared to other data mining and data analysis techniques MDS is growing increasingly popular because its mathematical basis is easier to understand and its results are easier to interpret (Fitzgerald & Hubert, 1987).

Permap is an interactive computer program. It offers both metric (ratio and interval) and nonmetric (ordinal, ratio + bounds, interval + bounds) MDS techniques. It solves problems in up to eight dimensional space and allows boundary conditions to be imposed on the solution. In the technical jargon, Permap treats "weighted, incomplete, one-mode, two-way" or "weighted, incomplete, two-mode, two-way" data sets. Other jargon would say it handles weighted, symmetric, incomplete, triangular or rectangular data sets. The word “weighted” means each data point can have its own multiplier that reflects in some way the importance or reliability of the point. The word “symmetric” means that Permap assumes that the (i, j) proximity value equals the (j, i) proximity value, and “incomplete” means that it can handle missing data. The one-mode, two-way and square references indicate that Permap can analyze a matrix of proximity information between several objects, and the two-mode, two-way and rectangular references means it can analyze objects each of which are specified by an array of attributes.

Permap can treat up to 1000 objects at a time (but see cautions in Section 11) and each object can have up to 100 attributes. It is easy to use, Windows PC-based, visually oriented, and allows real-time interaction with the analysis. It has been designed to have an intuitive interface and it avoids many of the arcane alternatives that are seen in the research literature but are never used in practice.

1.2 MDS Maps

Perceptual maps are sometimes called product maps, sociograms, sociometric maps, psychometric maps, stimulus-response diagrams, relationship maps, concept maps, etc. A perceptual map is a piece of paper, or any plane, with symbols on it that convey information about perceived relationships between the objects represented by the symbols.

What is the difference between a perceptual map and any ordinary map? Not much, although some people make a big deal about it. Usually, a perceptual map is taken to be a map that involves object-to-object relationships that are not amenable to simple, physical measurement.

What is an object? Objects can be anything. They can be stimuli, constructs, artifacts, characteristics, traits, people, companies, bones, arrowheads, words, discussion topics, and so forth. Anything that you want to study can be an object. If you are interested in how certain objects relate to each other, and if you would like to present these relationships in the form of a map, then MDS is the technique you need.

The MDS algorithm uses object-to-object proximity information to construct the map. What is a proximity? A proximity is some measure of likeness or nearness, or difference or distance, between objects. It can be either a similarity (called a resemblance in some disciplines) or a dissimilarity. If the proximity value gets larger when objects become more alike or closer in some sense, then the proximity is a similarity. If the opposite is the case, the proximity is a dissimilarity.

Proximity values can be calculated, measured, or just assigned based on someone's best judgment. If calculated, they typically are based on some mathematical measure of association (correlation, distance, interaction, relatedness, dependence, confusability, joint or conditional probability, pilesort counts, and so forth) operating on a set of attributes.

What is an attribute? An attribute is some aspect of an object. It may be called a factor, characteristic, trait, property, component, quantity, variable, dimension (not a good choice in MDS work, but occasionally seen), parameter, and so forth. The attributes should be presented in a form where each is normalized (standardized) to some kind of range or standard deviation, but Permap can do the normalizing internally if so desired. An attribute in one study may be an object in another study. It is all a matter of perspective and interest.

In this manual we present all formulas in terms of dissimilarities and refer to dissimilarities rather than proximities or similarities, but the choice is arbitrary. Typeset documents usually represent dissimilarities by the Greek del, δ_{ij}, where the indices i and j indicate that it is the proximity between object i and object j. However, the Greek symbols do not translate correctly in some MS Word versions configured for languages other than English, so here we use D_{ij} to represent a dissimilarity.

Ideally, the dissimilarity of each pair of objects is known. However, because MDS is a very robust procedure a few randomly missing values are of little consequence. In fact, except when using very small data sets, MDS is exceedingly good at yielding reproducible results even after many data points have been eliminated.

Dissimilarities do not have to be symmetrical. That is, in general there is no reason that the dissimilarity D_{ij} must always equal D_{ji} (note the reversal in subscripts). In fact, it is fairly easy to think of cases where this might be the case, such as where D_{ij} represents the driving distance between two stores (objects) and your town has one-way streets. In this case, your distance (dissimilarity) matrix will be asymmetrical. Fortunately, asymmetrical dissimilarity matrices are not too common in applied MDS work. In those cases where you do have asymmetrical dissimilarities, it is up to you to justify using the average of the two halves of the square matrix, ignoring one of the halves, or doing separate MDS analyses on each half and then somehow combining the results. All of these options are "right" for certain circumstances (see Young & Hamer, 1987, or almost any MDS text). Permap does not assist with resolving questions involving asymmetrical dissimilarity matrices.

Permap requires that dissimilarity values be nonnegative. (As a quick aside, Pearson correlation coefficients can be used, but they have to be shifted to the 0 to 1 range to keep all dissimilarities nonnegative. There are two ways of doing this. See Section 6.2 for details.) Also, the diagonal of a dissimilarity matrix must hold zeros. If similarities are used, the diagonal of the similarity matrix must hold identical values which are not exceeded elsewhere in the matrix.

If you are new to MDS, you might benefit from the books by Cox and Cox (2000), Davison (1992), and Borg and Groenen (2005). The Cox and Cox text is the most mathematically orientated, Davison's is the most applied, and Borg and Groenen's falls somewhere in between the other two. To varying degrees, all three texts cover theory, application, and historical development. Young and Hammer (1987) is the best in terms of explaining the historical roots of MDS.

2.0 GETTING STARTED

2.1 System Requirements

Permap will run on any contemporary personal computer using a Windows operating system. It is not particularly demanding in terms of computer speed, but if the problem is large, say more than 100 objects, then at least a 1-GHz computer is recommended. Problem solution time is often controlled as much by the computer's available RAM (random access memory) as it is by the computer's processor speed. With the XP or 2000 Windows operating systems you should have at least 512 MB and preferably 1 GB of RAM. Vista needs 1 GB or more of RAM.

2.2 Installation

Obtain a copy of the file Permap.zip. This file is available on the Internet at www.ucs.ull.edu/~rbh8900/permap.html. After downloading the Permap.zip file, open it using any of numerous decompression utilities. Usually you will have a zip/unzip program included as part of Microsoft suite of programs. If not, one such utility is ZipGenius available at http://www.zipgenius.it/index_eng.htm. ZipGenius is a free program that is particularly user friendly, downloads fast, installs easily, and can be run with no training.

After unzipping Permap, you will have Permap.exe, some Windows library files (ocx and dll extensions), some example files, a default-values file, and a WAV file. You might want to put a shortcut to Permap.exe on your desktop. Right click on Permap.exe (in whatever directory you chose to put the unzipped files) and drag it to the desktop. Choose "Create Shortcut Here" when you release the right mouse button. Permap is now ready to use. Permap can be uninstalled simply by erasing the directory you used to hold the Permap files because no Registry entries have been.

Start the program by double clicking the Permap shortcut, by double clicking on Permap.exe in the directory where you put Permap, or by entering its location and name in the Run input box's "command line" which is accessed via the Start button.

Permap can be controlled by a mouse or keyboard. The alt-key gets you to and from the menu bar, the tab-key shifts between controls, and the enter-key activates a control.

2.3 Data Input

Data are entered from a text file (i.e., a file stored in ASCII or ANSI format). Any modern word processor can save information in these generic formats. For instance, if you use WordPerfect, use File/Save As and choose ASCII DOS Text or ANSI Windows Text from the File Type box at the bottom of the screen. If you use MS Word, use File/Save As and choose Text Only or MS-DOS Text from the Save As Type box at the bottom of the screen. Alternatively, it is often faster and easier to use Notepad, the simple text editor that comes with all Windows operating systems. Notepad is designed for quick entry of short segments of unformatted text.

Permap's data files are based on freeform data entry. This means that keyword identifiers announce the presence of various data elements and that these data types can be present in the file in any order. Comment lines can be placed freely throughout the data file as long as they are not placed between a keyword and its following data. All optional information is covered by default values. This means that if you choose not to use weights then they need not be mentioned in the data file.

Here is a very simple data set that assumes your data are in the form of dissimilarities. All the sentences to the right are for explanation purposes and are not part of the actual data file.

```
NOBJECTS=6                  Gives the number of objects in the analysis.

DISSIMILARITYLIST           Announces that dissimilarity values follow.
0                           All values must be nonnegative and diagonal values
.21, 0                      must be zero.  The data can be separated with space(s),
.59, .68, 0                 a comma, or both.  DISSIMILARITYLIST is all one
.74, .79, .2, 0             word and capitalization is optional.
.88, .8, .24, .25, 0        Missing entries are shown by "NA" or "na."
.11, .1, .66, .7, .89, 0    It is acceptable to name the objects.  See Appendix II.
```

Dissimilarity data can be in either a lower-left half-matrix, as shown above, or in a whole matrix format. If a whole square matrix is entered, the upper-right triangle is ignored. Entering a square matrix is allowed simply to facilitate data interchange with other programs such as Excel.

You may wonder why it is necessary to enter the zeros of the diagonal when dissimilarity data are involved. After all, if the data are dissimilarities then these diagonal values must be zero by definition. The answer is that it is not really necessary. Permap will enter the missing diagonal values for you if you don't enter them. However, experience shows that beginning MDS users often get similarities and dissimilarities mixed up unless they are encouraged to enter the zeros whereupon they usually get it correct right from the start. Unfortunately, a previously used convention of requiring that the diagonal values be in the data file was not much admired by Permap users. In fact, it has been the source of so much grumbling that, starting with version 9.7 Permap was modified to enter the missing dissimilarity zeros if need be. But remember, this does not apply to similarity data. You will get an error message if the similarity diagonal values are missing.

If your proximity information is in the form of similarities instead of dissimilarities, then replace the keyword DISSIMILARITYLIST with SIMILARITYLIST and be sure that the diagonal values are all equal and are not exceeded by any other similarity value. There is no space before the "LIST" part of the keyword and capitalization is not important except for readability considerations.

If you want to use proximity data from a spreadsheet then first save the spreadsheet in ASCII format. For instance, if using Excel, use File/Save As and choose either Text or CSV from the Save As Type box at the bottom of the screen. Then, load the saved file in a text editor and enter the keywords (e.g., DISSIMILARITYLIST), titles, and so forth. Save the edited file in ASCII text format with a "txt" or "dat" extension, and use this new file as input to Permap.

Instead of entering proximity data, Permap will accept attribute data. In this case, each object is represented by a set of attribute values and the proximities between objects are calculated from the attribute data using any of several standard relationships. If you want to use attributes to generate the proximities then use the following format (only the left-justified entries are needed).

NOBJECTS=6	Gives the number of objects in the analysis.
NATTRIBUTES=3	Gives the number of attribute values for each object.
ATTRIBUTELIST	Announces that attribute values follow.
1, 1, 2	Any range can be used for the data.
1, 2, 2	The data can be separated with space(s), a comma,
3, 3, 2	or both. ATTRIBUTELIST is all one word and
3, 4, 0	capitalization is optional. All data entries must be numbers.
3, NA, 0	Missing entries are shown by "NA" or "na."
1, 6, NA	It is acceptable to name the objects. See Appendix II.

You may want to put a title on the map. You do this with the TITLE or MESSAGE command somewhere (anywhere but between a keyword and its associated data) in the data file. For example, enter this.

TITLE= Whatever follows "=", such as this line, is transferred to the top center of the output page.

A complete list of keywords is shown in Appendix I. Appendix II shows an annotated data input file. EXAMPLE_A.txt shows a typical MDS problem expressed in terms of similarities, and EXAMPLE_B.txt shows one using attributes. The EXAMPLE_A.txt and EXAMPLE_B.txt files, as well as several other example problem files, are included in the Permap.zip package and can be loaded by using the File/Load Data From a Text File or by using the F2 shortcut key.

2.4 Data Input, MDSX

To make it easier to use input files originally made for the MDSX family of programs (see http:// www.newmdsx.com/), Permap accepts certain alternate commands (keywords) to those shown in Appendix I.

The following table gives equivalencies between the MDSX commands and Permap commands. The translation between MDSX and Permap is difficult because MDSX is really a suite of programs, some of which treat advanced MDS topics that are not covered by Permap. Thus, there are some equivalencies that occur in special cases but are not listed below. Also, the following list ignores MDSX control commands that are not applicable to a real-time interactive environment. For example, input file commands COMPUTE, DIMENSIONS, ERROR LIMIT, FINISH, INPUT MEDIUM, NUMBERED, PLOT, PRINT, and PRINT DATA are not needed in an interactive environment because they can be issued (with mouse clicks) at any time by the program operator. Further, the MDSX keyword INPUT FORMAT is not needed because Permap accepts any numerical format.

Table 1
MDSX and Permap Input File Command Equivalencies

MDSX	**Permap**	**Details / Comments**
COMMENT	no equivalent	Any line that does not start with a Permap keyword is considered to be a comment.
RUN NAME	TITLE = or MESSAGE =	Permap’s title has an upper limit of 65 characters and the "=" is optional.
TASK NAME	SUBTITLE = or SUBMESSAGE =	Permap's subtitle has an upper limit of 98 characters and the "=" is optional.
# OF STIMULI	NOBJECTS =	MDSX's "#" can be "N" or "NO" and Permap's "=" is optional.
# OF VARIABLES (used in WOMBATS)	NATTRIBUTES =	MDSX's "#" can be "N" or "NO" and Permap's "=" is optional.

READ CONFIG	LOCATIONLIST	Follow with a list of X, Y pairs separated by a space or a comma.
PARAMETERS DISSIMILARITIES READ MATRIX	DISSIMILARITYLIST	MDSX's input is on two separate lines.
PARAMETERS SIMILARITIES READ MATRIX	SIMILARITYLIST	MDSX's input is on two separate lines.
PARAMETERS ATTRIBUTES READ MATRIX	ATTRIBUTELIST	MDSX's input is on two separate lines.

3.0 MAIN SCREEN & PROGRAM CONTROL

This section gives a detailed description of Permap's main screen. It covers many fairly obvious aspects of operating Permap and can be skipped by users that are familiar with interactive computer programs that use a graphical interface.

The MDS map is presented in the center of the screen. Map identification information and the objective function value are given in the header section. A group of control buttons allow real-time manipulation of the most important problem parameters.

3.1 Header and Objective Function Value

The title can be up to 65 characters long and the subtitle can be up to 98 characters long (a variable font is used so the exact number of characters depends on the specific characters involved). Titles and subtitles are optional. Just below the header is a line that shows the latest value of the objective function. This value will converge to a minimum as the iterative solution process continues. If the Auto Repeat option is being used, then just below the current objective function value the best-found value is displayed, followed by the number of times it has been found, followed by the total number of times the problem has been solved for the given set of parameters. The number of times the best-found value has been found can depend on Auto Repeat's precision (convergence limit) setting. Please see Section 3.5.4, Automatic Controls, for more information on this point.

3.2 On-line Shortcut Controls

The left side of the main screen has a group of shortcut buttons that allow easy control of the basic analysis parameters or functions while the solution process is continuing. Each click on a shortcut button causes the selected parameter or function to rotate to the next setting. For instance, by clicking on the Distance shortcut button, you can cause Permap to measure map distances using the Euclidean measure, then the City Block measure, then the general Minkowski measure, and then back to the Euclidean. Holding down the shift key while clicking one of the shortcut buttons causes the rotation to reverse direction. Each of these shortcut buttons has an associated shortcut key for those that prefer using the keyboard instead of the mouse. See Section 3.6 for a list of the click and keystroke commands, or observe the underlined letter in each button's caption.

The shortcut controls duplicate features available through the menu system. The menus give more information about each parameter or function, and in some cases allow more options to be controlled, while the main screen's shortcut buttons provide easier real-time control. Note that the shortcuts only show alternatives that are consistent with the more detailed settings specified on the menu screens. That is, if you use the Attributes screen to specify that only ordinal level attribute-to-dissimilarity functions be used, then the Attributes shortcut button will only rotate to the Spearman and Guttman correlation coefficients because they are the only ordinal level attribute-to-dissimilarity functions built into Permap.

3.3 Parking Active Objects Control

The parking lots are on the right side of the main screen. Objects placed in the object's parking lot are removed from the active data set. This makes it easy to discover the influence of a particular object or attribute on the overall configuration. Double click an object to move it into its parking lot. Objects also can be dragged into the object parking lot. To return an object to the active set just drag it out of the object parking and to the position you want it to be when the analysis continues. Each time an object is parked or unparked, the map readjusts to represent the new active data set.

3.4 Parking Active Attributes Control

If your data file contains attribute information, just below the functional shortcut buttons will be a box that shows the "active attributes." These attributes are used to calculate the dissimilarities by using any of several built-in attribute-to-dissimilarity functions.

Double clicking on an attribute moves it to the attribute parking lot and the dissimilarities are recalculated using only the remaining active attributes. A parked (inactive) attribute can be brought back into the active set by double clicking on it in the attribute parking lot.

3.5 Execution Control

A set of execution control buttons appears just below the map in the center of the screen. Execution of the program can be started, stopped, continued, jiggled, subjected to field movements, automatically stopped, or automatically started by using these controls.

3.5.1 Start, Stop, Continue

When Start is clicked, the objects are placed at random positions on the map and the iterative solution procedure is commenced. The Stop button stops the iteration and displays the object labels if the labeling option has been selected (see Section 4.3.2). If the solution has converged to a stationary value of the objective function but a stop signal has not been given, then the ellipses following the objective function value continue to blink. If you want to resume iteration after a stop command has been sent (either from the Stop button, from Auto Stop, or from a click on an open area of the map) without using new starting positions, then click the Continue button.

You might notice what appears to be a problem with the Continue command. Say you are using dissimilarity data that describe a perfect circle, e.g., you are using the first of the Special Cases built into Permap (see Section 9.0). After finding the solution you drag an object out of the circle and into the object parking lot. Then you click Continue and you expect nothing to change, but you find that all of the objects move a little. Here is what is happening. Your expectation that the map of all the remaining objects should not change is correct, and in fact their relative positions do not change. The overall movement is because Permap maintains the map's centroid at (0, 0), and unless the object you removed was located precisely at (0, 0) the new centroid, i.e. the centroid after recalculation without the parked object, will be slightly different. So, as soon as Continue is clicked, Permap moves all objects just a bit to return the centroid to (0, 0). MDS maps that differ only by translation, rotation, or reflection are considered to be identical, so the map of the remaining objects does not "change" even though it moves.

3.5.2 Jiggle

Clicking the Jiggle button causes the objects to be moved to new positions that are randomly perturbed from their existing positions. This is called "jiggling the solution." A simple Jiggle click randomly displaces the objects a distance that averages 5% of the map diameter. If the shift key is depressed while clicking Jiggle, the average displacement is 10% of the map diameter. Using the control or alt keys causes average displacements of 20% and 40%, respectively.

The Jiggle command contributes to understanding the nature of the solution. For instance, it can provide information on the "depth of the potential well" that surrounds the object positions. By slightly displacing the objects and watching the speed and directness of the reconvergence you can judge if the displayed map is clearly better than all neighboring configurations, or just one of several "almost as good" object placements. You may be surprised to find that some local minima can be very stable, even when subjected to large perturbations.

3.5.3 Field Movements (Mirror, Rotate, Move, Zoom)

Occasionally you will want to control the final orientation of a map in order to make a simple comparison to previous results, or you might want to expand a map to inspect a small, congested, area. These needs can be satisfied by mirroring, rotating, moving, or zooming in. These operations are known as "field movements" because the chosen operation applies to the whole field of objects. The field movement controls are activated by clicking the Field button or right clicking the mouse on an open area. Alternatively, you can press the F-key. To use one of the four field movement commands, first select the desired type and then use the mouse to control the execution of the movement. If Mirror is chosen, then clicking near an axis will cause the map to be mirrored about that axis. If Rotate is chosen, then dragging the mouse about the center of the map will cause the object set to rotate about the center. If Move is chosen, then dragging the mouse in any direction will cause the object set to move in that direction. Finally, if Zoom is chosen, then dragging the mouse away from the center of the map will cause the object set to expand, and vice versa. The results of all of these movements persist after the field controls are closed.

Remember that if a MDS solution is mirrored it still is a valid solution. This is true for any type of MDS, badness measure, distance measure, and weighting scheme. The validity of a solution is similarly invariant to all rotations, but only if the Euclidean distance measure is used. If City Block or any Minkowski distance measure with p not equal to 2 is used, then only 90 degree rotations do not change the optimality of a solution. This can be demonstrated by solving a problem using the City Block distance measure, then rotating the solution about 45 degrees, and then continuing the solution. The map will immediately rotate ±45 degrees to again reach optimality.

3.5.4 Automatic Controls

The Auto Repeat and Auto Stop check boxes control whether or not the solution process automatically stops, and then possibly automatically restarts, after a certain convergence limit (as set using the Convergence menu) is reached. These controls facilitate finding numerous solutions, each starting from a new set of random positions, to be sure that a global minimum has been found.

If you are having trouble finding the global minima you might like to let the computer run all night and have it record the best solution it finds. To do this, check Auto Repeat and Auto Stop, and just let the program run. The best-found value is displayed in brackets just below the current objective function value. Next to the best-found value is the number of times that the best-found value has been found, followed by the number of times the problem has been solved using the current set of parameters.

The number of times the best-found value has been found is influenced by Auto Repeat's precision (i.e., convergence limit) setting. When the convergence setting is on "low precision" all values within 0.0005 of the best-found value are counted, when it is on "normal precision" this tolerance is 0.00005, and when it is on "high precision" it is 0.000005. If your goal is to quickly scan many cases, then use a low precision setting. However, if there are several mappings (local minima) that have objective function values that are close together it will be difficult to determine how many times the best-found mapping actually has been found. In these cases it is important to use the high precision setting. Also, some ordinal MDS solutions tend to pulsate and transient objective function values can be caught when the precision value is too low. So, when finalizing your work it is important to use the high precision setting. If in doubt, reload (see below) the best-found solution and let Permap work on it for an extended period of time. If the objective function value decreases, then you caught a transient configuration with the automatic stop feature.

The complete details of the best-found solution are stored in the file PermapBestSolnFound.txt. The PermapBestSolnFound.txt file can be reloaded via the File menu to continue your analysis at a later date. A shortcut to reloading the last best-found file is to press F7 and then select SaveExitReload from the menu bar.

You will find that many "real" MDS problems have several local minima and need the Auto Restart feature to capture the global minima. The Ekman (1954) color comparison data that is provided in EXAMPLE_A.txt is typically susceptible (correct answer: using ratio MDS, Euclidean distance, two dimensions, and normalized Stress, the objective function value is 0.01721). The topic of finding global versus local minima leads to the topic of using certain "smart" starting points to help the solution procedure converge to a global minima. See Section 11.0, Solution Difficulties, for more information on this important topic.

3.6 Click & Keystroke Commands

Experienced users often want the fastest possible ways of controlling program execution even if the commands are not particularly intuitive. Here are some such control techniques using the mouse, keystrokes, or combination keystrokes. All of these apply while the main screen has the focus. The familiar Windows cut, paste, and select commands still work in an edit window, even though their control keys have been usurped when the main screen is active.

Mouse shortcut controls:

Clicking an open area toggles between the Stop and Continue buttons.
Double clicking an open area is equivalent to clicking the Start button.
Double clicking an object parks the object.
Double clicking an attribute number parks the attribute.
Right clicking an open area reveals the field movement’s controls.
Control clicking an object toggles between locking and unlocking the object's position.

Keystroke shortcut controls:

F1 opens the help file
F2 opens the file selection dialog box
F3 opens the special cases screen
F4 opens the active data file
F5 opens the last-saved solution data file
F6 opens the last-saved short solution data file
F7 opens the best-saved solution data file (saved by Auto Repeat feature)

A-key changes the Attributes function
B-key changes the Badness function
C-key clicks Continue
D-key changes the Distance function
F-key reveals the Field Movements controls
E-key changes the level of D_{ij} Precision
J-key clicks Jiggle
M-key changes the Mapping Weights function
N-key changes the number of Dimensions in the solution
P-key clicks Stop
S-key clicks Start
T-key changes the MDS Analysis Type function
X-key clicks Exit
Y-key changes the Boundary size

esc-key clicks Exit

Alternate-key shortcut controls:
alt-A opens the Analysis Parameters menu
alt-B opens the Attributes-to-Dissimilarities screen
alt-C opens the Special Cases screen (same as F3)
alt-E opens the Edit menu
alt-F opens the File menu
alt-H opens the Help menu
alt-N opens the Convergence screen
alt-P opens the Map Evaluation menu
alt-V opens the View screen

Control-key shortcut controls:
ctrl-A opens the Attribute Evaluation screen
ctrl-B opens the Objective Functions selection screen
ctrl-D saves the existing analysis settings to file PermapDefaults.dft
ctrl-L opens the Links Evaluation screen
ctrl-M opens the Mapping Weights selection screen
ctrl-N opens the Dimensions and Boundary selection screen
ctrl-O opens the Objects Evaluation screen
ctrl-P prints the screen using the Windows PrintScreen command
ctrl-S saves the input data and the solution to file PermapSolutionLongFmt.txt
ctrl-T opens the MDS Types and Error Bounds selection screen
ctrl-Y changes the Boundary size at a slower rate than the Y-key

3.7 Locking Object Positions

You may want to fix the position of one or more objects on the map in order to make an in-depth study of sub-optimal configurations. This situation could occur, for instance, if you are absolutely confident of the relative positions of certain major objects, but are unsure whether or not more arrangements of the minor objects are reasonable.

Locking an object's position can be done by holding down the control key while clicking the object. Unlocking an object is done similarly, that is, hold down the control key and click on the locked object. Locked objects can be dragged to a new positions. The field movement options, mirror, rotate, move, and zoom can move locked objects because in these cases the whole object set is being modified and relative positions do not change.

Even with some of the geometrically simple cases stored in Permap's special cases library it is common to discover unexpected solutions when one or more objects are locked in place. These special-situation solutions should not be dismissed out of hand. One should first ask "could something in the environment cause one or more objects to be less mobile than the others." If so, sub-optimal solutions may represent a physical reality.

4.0 FILE, EDIT, VIEW

This section describes the use of Permap's system of menus. As with Section 3.0, it can be skipped by users that are familiar with Window-based interactive computer programs.

4.1 File Menu

The File submenus provide several ways to load, save, or print data. These Input/Output submenus use the format common to all Windows-compatible applications. The shortcut key for this option is alt-F.

4.1.1 Load Data from Existing Files

Data are entered in either of two ways. Data can be read from a text file or entered automatically by selecting any of several special case data sets (see Section 9.0). The format of the data file is described in Section 2.3, Appendix I, and Appendix II. In addition to using the menus, the F2 function key provides a shortcut that automatically opens the input file dialog box. The F4 function key provides a shortcut that automatically opens the active data file so that it can be reviewed or edited. The F5, F6, and F7 function keys allow reloading the three kinds of "saved solution" files.

4.1.2 Load Data From a Special Case

The Special Cases screen can be opened using the F3 shortcut, or by following the submenu File/Load Data From a Special Case. Due to requests from the "one click mouse people," it has been duplicated as a top level Special Cases menu. Thus, it also can be opened by the alt-C keystroke or by one click to the menu tab. Please refer to Section 9.0 for its operation.

4.1.3 Enter New Data

If you want to define a new problem, often it is easier to call up an old data file, modify it, and then use Save As (described in the next section) to save it with a new name. This approach can be easier because it provides a "template" to guide your new entries. However, sometimes you will be pasting data from some other media, and it will be easier to start with a clean slate. To start from scratch, use File/Load Data From a Text File, or ctrl-N, and a text box will open.

The default file name for the new file is "NewProblem.txt." You can type entries, cut, paste, copy, etc. in the text box. To avoid overwriting valuable data the new data is not automatically saved upon exiting this window, so be sure to explicitly save it when finished. Normally, you will use the Save As option to give it some name other than NewProblem.txt so that the original file will not be erased if a later Permap user decides to use this option.

4.1.4 Save Results & Settings

Solution results and analysis settings can be saved to a file. There are four options.

The first option, File/Save Solution (Long Format), preserves all input values needed to completely specify the problem and completely specify its solution. Because it includes printing the distance and dissimilarity matrices this file can be very large when the number of objects is large. The information is written to the text file PermapSolutionLongFmt.txt. If PermapSolutionLongFmt.txt already exists, it is overwritten. This solution file has keyword headers suitable for restarting the analysis at the state existing when the save command was given. That is, the output file PermapSolutionLongFmt.txt can be used as an input file. The shortcut key for this option is ctrl-S.

The second option, File/Save Solution As … (Long Format), does the same as the previous option except that it allows the specification of a different file name.

The third option, File/Save Solution (Short Format), writes out an abbreviated file that fully specifies the solution, but does not repeat all the input data that define the problem. The file name is PermapSolutionShortFmt.txt and any other copy of PermapSolutionShortFmt.txt is overwritten. The shortcut key for this option is ctrl-R.

The fourth option, File/Save Current Settings As Defaults, saves all existing settings to the defaults file, PermapDefaults.dft. This allows restarting Permap with all of the existing settings. Thus, once you decide on a set of preferred settings these settings need not be reset each time Permap is started. The shortcut key for this option is ctrl-D.

If you want to read or edit any of these saved solution files, including the one that is automatically saved by the Auto Repeat feature, use the Edit menu or the F5, F6, or F7 function keys.

4.1.5 Reload Original Default Settings

When a program has as many parameters as Permap has it is easy to get "lost" and not be able to return to previously used conditions. Most often, this occurs because some obscure setting has been changed and forgotten about. Selecting File/Reload Original Default Settings will cause Permap to reset all parameter settings to commonly used settings and write out a new default file with these settings. If you are sharing a Permap installation with others, it may be advisable to start each session by clicking File/Reload Original Default Settings.

4.1.6 Printing Results

Most analysis time is spent examining maps drawn on the computer screen. However, at some point a paper copy is usually needed. There are two approaches to getting the MDS map onto a piece of paper.

The first approach is to use the information provided by Permap to construct and print a map using a graphics application. This can be done, for instance, by saving the object's coordinates using one of the File/Save commands discussed in Section 4.1.4, and then importing the saved object coordinates into a graphics or spreadsheet application to prepare a presentation-quality map.

The second approach is to save the map to the Windows clipboard by using the PrtScn key to capture the entire screen, or alt-PrtScn to capture just the active window, and then open Paint or another graphics application and paste the clipboard contents. After modifying the map as desired, it can be printed from within the graphics application.

The third approach is to have the Windows operating system print the screen directly to your attached printer. This can be done using "File/Print Screen (Std. Windows)" or the shortcut key ctrl-P. Windows does not allow control over placement (output will be in the upper left of the printer's page) or size (the output will be truncated if it exceeds the dimensions of the paper). You will probably want to change the orientation of your printed page to “landscape.” Do this with the following sequence: Start Button → Settings → Control panel → Printers and Faxes → (select the printer you will be using) → File → Printing Preferences → Landscape.

4.2 Edit Menu

The usual Windows editing keystrokes work when an editing window is open. That is, ctrl-A selects all, ctrl-C copies the selected content to the clipboard, ctrl-X cuts the selected content and copies it to the clipboard, and ctrl-V pastes the contents of the clipboard at the position of the cursor. Note that these shortcut keys work as described only within an editing window. For instance, if ctrl-A is pressed when the main screen is active it opens the Attributes Evaluation screen, but when pressed inside an editing window it is the traditional "Select All" command.

The following Edit screens are most easily opened using the F1, F4, F5, F6, and F7 shortcut keys. Because you may forget which function key goes with which edit screen, just like we do, these shortcuts have been set up to return you to the main screen if they are pressed twice. That is, pressing F1, etc., twice results in no net change. This makes it easier to "just hit something" when you want to edit one of the three main data files and you don't remember which function key is the correct one.

4.2.1 Edit Active Data File

You may edit your data file (or any other ASCII file for that matter) from inside Permap. Enter the Edit menu and choose the Edit Active Data File selection. It will open the data file and allow you to use all the usual Windows keystrokes. If you want to work on another file, use the File option from within the Edit menu. The shortcut key for this option is F4.

If you change the active data file, the changes do not automatically take affect for the problem that is loaded. You must first save the modified file (ctrl-S), exit the Edit menu, and reload the file that you just saved to have Permap analyze the data.

Alternatively, you can use the SaveExitReload menu item to do this series of steps automatically.

4.2.2 Edit Help File

The Edit menu also has the option of opening the help file (PermapHelp.txt). This file contains several sections of this manual that contain operational details. It is basically just a reminder list. It is made available through the Edit menu so that you can make notes in the help file, and then save the file for future reference. Thus, with time the help file becomes your personalized help file. The shortcut key for this option is F1.

4.2.3 Edit Solution Files

The Edit menu allows easy access to the problem solution files (PermapSolutionLongFmt.txt, PermapSolutionShortFmt.txt, PermapBestSolnFound.txt), i.e., the files you formed by using ctrl-S, ctrl-R, or the File/Save Solution menus, or the file that is automatically saved by the Auto Repeat feature. Often you might want to consult one of these files to find some numerical value of a parameter for a solution that is under study. Or, you might want to change some particular parameter's value and then watch what happens when the problem is rerun. The shortcut keys for these options are F5, F6, and F7. As with the Active Data File case, changes to PermapSolutionLongFmt.txt and PermapBestSolnFound.txt can be automatically saved and reloaded to be the active data file by using the SaveExitReload menu item. This cannot be done with the PermapSolutionShortFmt.txt file because it does not have all the information necessary to fully describe the problem.

4.3 View Menu

Selecting the View menu opens the View Screen that lets you control the look of the MDS map. The shortcut key for this option is alt-V.

4.3.1 Object Representation

You can customize the appearance of your map by selecting the object size, object border width, and object border color. Each option is explained in a text box at the right side of the View screen and a demonstration map at the bottom-right of the screen illustrates the current settings. When you are analyzing large problems it is beneficial to reduce the circle size to minimize overlaps and thus clarify the map.

4.3.2 Object Identification

User-supplied names, preferably just a few characters each, can be used to identify the objects once the analysis is stopped. Or, the objects can be automatically numbered in the order they are entered in the data file. As useful as these options are for small problems, maps with many objects quickly become congested. For large problems, the naming option should be turned off by selecting the "none" option.

Object names are displayed only when the solution (process iteration) has stopped. You can stop the process by using the Stop key. Alternatively, you can toggle between the Stop and Continue states by a single mouse click on an open space on the map.

4.3.3 Unit Circle and X-Y Axes

Some users prefer to show a unit circle on map. All metric dissimilarity matrices generate object positions that lie within this circle. You turn on this feature with the first entry under View/Mapping Space.

The second Mapping Space entry controls the display of an X-Y coordinate system. The X-Y coordinate system described here is the internal system that is used to make the calculations and place the objects on the map. While some users like to show these coordinates, they are easily misunderstood and probably should not be shown unless you are making a City Block or some other non-Euclidean distance analysis. These coordinates are not related to the interpretative coordinates that the researcher assigns to the map.

5.0 ANALYSIS PARAMETERS

A complete MDS analysis description requires specifying the type of MDS analysis, the objective function (badness and distance measures), mapping weights, solution dimensionality, boundary conditions, and the nature of the attributes being used (if any). The Analysis Parameters set of submenus let you specify all of these aspects and assists in testing the affect of changing dissimilarity precision via the Precision control button. Open this menu with alt-A or by clicking on the Analysis Parameters menu.

5.1 MDS Analysis Types

The first option in the top box on the left side of the MDS Types and Error Bounds selection screen allows selection of the type of MDS analysis to be made. This selection also can be made by clicking the MDS TYPE shortcut button or by opening the MDS Types and Error Bounds submenu using ctrl-T.

Permap can make ratio, interval, or ordinal analyses, as well as ratio and interval analyses with error bounds. After providing some general comments on MDS terminology, each of these methods is described in detail. The specific formulas used to match analysis type to the objective function are given in Section 5.2.

5.1.1 Metric and Nonmetric Terminology

Traditionally, MDS analyses are classified as metric or nonmetric. These terms are not entirely satisfactory because of their imprecision, but they persist. More precisely, one should classify MDS analyses based on the relationship between the map's inter-object distances d_{ij} and the dissimilarity D_{ij} values. There should be at least three major types of MDS analysis. First, if the algorithm seeks to find d_{ij} such that $d_{ij} / D_{ij} = k_1$, where k_1 is an arbitrary constant that determines the size of the map, then the MDS algorithm is at the ratio level. Second, if the algorithm seeks to find d_{ij} such that (d_{ij} -

d_{kl})/ (D_{ij} - D_{kl}) = k_1, then the MDS algorithm is at the interval level. Third, if the algorithm seeks to find d_{ij} such that $D_{ij} < D_{kl}$ implies $d_{ij} < d_{kl}$, then the MDS algorithm is at the ordinal (rank order) level. Based on these considerations, an MDS analysis should be classified as ratio, interval, or ordinal instead of metric or nonmetric.

Ratio and interval levels of information are two examples of metric information. Ordinal information is an example of nonmetric information. The word "metric" harks back to "measurement" and signifies that measured distances on the map are linearly related to the D_{ij} values. If you simply ignore the important difference between ratio and interval levels of information, you arrive at the historical terminology where all MDS analyses are classified as metric or nonmetric

5.1.2 MDS and Error Bounds

Permap provides two new nonmetric kinds of analysis that involve applying error bounds to ratio and interval level data. Thus, Permap makes traditional ratio (metric), interval (metric), and ordinal (nonmetric) MDS analyses, and adds ratio + bounds (nonmetric) and interval + bounds (nonmetric) analyses. Before presenting precise definitions of the various types of MDS maps that Permap can make it is important to understand the role of error bounds and why the traditional forms of MDS have ignored the error bounds approach.

Every experimental measurement has some associated uncertainty. However, if the MDS computer program does not have the ability to use the uncertainty information, you must either make the invalid assumption that the dissimilarities should exactly control the distances between objects (traditional metric analyses), or make the unrealistic assumption that the dissimilarities are so poorly known that only their rank order can be trusted (traditional nonmetric analyses). You should not be forced to choose between these two extremes.

Metric MDS is based on the reasonable "first order approximation" that the input data are linear in the construct of interest and are errorless. In the physical sciences the second step usually is to relax the assumption of errorless data and to introduce error bounds. However, the development of MDS did not follow the usual path. Early MDS critics were fixated on the point that behavioral science measurements often are very imprecise. The result was that metric MDS was not modified to accept error bounds. Instead, ordinal MDS was developed and it essentially took over MDS. There are even comments in the early MDS literature referring to the "near magical" nature of ordinal MDS. The ordinal advocates, in effect, chose to discard useful information simply because not all of the information was at a high level of reliability.

Here is an example of what happened. Say you have measurements, or estimates, yielding dissimilarities of 1, 1.05, 1.1, 8, 9, 10, and 19. It is quite possible that your experimental uncertainty is such that you cannot prove that the first three cases are truly different from each other. Also, it is quite possible that while you are sure that cases four through six are properly ordered, you are not at all sure that there is exactly one unit of difference between them. As a result, ordinal advocates argue that the analysis should proceed as if the dissimilarities maintain the rank order of the original data, and that is all. Their analyses would use pseudo D_{ij} of 1, 2, 3, 4, 5, 6, and 7 in place of those shown above. Certainly the ordinal advocates have valid concerns. D_{ij} data are often fraught with imprecision. But, just because some relationships are uncertain it does not follow that the whole data set should be treated as ordinal. There is little doubt that you know the seventh data point is far higher than just one step above the sixth, and that the differences in the first group of three clearly are not the same as the differences in the second group of three.

The correct way to approach most data sets is to ignore the old metric vs. nonmetric arguments and simply face the fact that all experimental data have error. Doing so suggests that every entry should have a best-estimate value and associated error bounds. There are many ways to specify error bounds, but the differences are relatively unimportant compared to the importance of avoiding the mistake of degrading to an ordinal analysis when it is not necessary to do so. If, for example, all of the dissimilarities mentioned above have associated uncertainties of ± 0.2, and an analysis is made taking these error bounds into considerations, then the first three dissimilarities points will be treated as being essentially the same. And, the far-different last dissimilarity (i.e., 19) will still be recognized as being far different from the others. Nowhere do we claim that we know the true values of each dissimilarity, but we do preserve the significant differences that we do know to be correct.

To restate the forgoing argument in less quantitative terms, we can say that while our measurements of some quantity may be sufficiently inaccurate that we cannot assign precise values we may be able to say much more than just what rank order the quantities have. Sometimes we can justifiably say that one value is "a little greater than," "much greater than," or "very much greater than," another value instead of just saying one is greater than the other.

5.1.3 Procedural Details for Each Analysis Type

In the following sections each type of MDS analysis is defined in terms of its quantitative relationships. The formulas behind these definitions are given in Section 5.2.

5.1.3.1 Ratio

A ratio MDS algorithm attempts to find a set of d_{ij} that approximately satisfies the relationship $d_{ij} / D_{ij} = k_1$, where k_1 is an arbitrary constant that determines the size of the map. This is the same thing as saying the goal is to find a set of d_{ij} such that a linear relationship $d_{ij} = k_1 D_{ij}$ is fit as well as possible. In Permap, k_1 is always unity because the dissimilarities or similarities are scaled to a maximum of one immediately upon data entry. Therefore, k_1 drops out of Permap's internal equations, and is omitted from the formulas outside of Section 5.1.

5.1.3.2 Ratio + Bounds

A ratio + bounds analysis attempts to find d_{ij} that satisfy $d_{ij} / D_{ij} = k_1$, but the objective function is not penalized as long as d_{ij} is within the range $k_1 (D_{ij} \pm e_{ij})$ where e_{ij} represents the error or uncertainty in D_{ij}. Permap can handle two types of error terms.

The first option in the second box on the left side of the Analysis Type and Error Bounds selection screen is the one that most new users should start with. It simply assumes that each D_{ij} term is subject to the same uncertainty $\pm e_{ij}$. While this approximation may not be precisely correct, it is often a good starting point. By clicking on the up or down arrows, the bounds can be set anywhere up from 0 to1. If the error bounds are set to 0, then this option becomes identical to the traditional ratio analysis and is eliminated from the rotation when the MDS TYPE shortcut button is used.

The second option is best for those situations where the uncertainty is proportional to the size of the factor being measured. This is often the case in physical measurements. This option assigns uncertainties of plus or minus some constant times D_{ij}. By clicking on the up or down arrows, the percentage error bounds can be set anywhere from 0% to plus or minus 100% of D_{ij}. If the error bounds are set to 0%, then this option becomes identical to the traditional ratio analysis and is eliminated from the rotation when the MDS TYPE shortcut button is used.

5.1.3.3 Interval

An interval MDS algorithm attempts to find a set of d_{ij} that at least approximately satisfies the relationship $(d_{ij} - d_{kl})/ (D_{ij} - D_{kl}) = k_1$, where k_1 is an arbitrary constant that determines the size of the map. This is the same thing as saying that the goal is to find d_{ij} that at least approximately satisfies the relationship a $d_{ij} = IC + k_1 D_{ij}$, where IC is the Interval Constant. As was mentioned in Section 5.1.3.1, in Permap, k_1 is always unity. Thus, it is omitted from further discussion and is omitted from the formulas outside of Section 5.1. When making an interval MDS analysis, Permap finds the value of IC that minimizes the objective function and displays it in the caption of the MDS TYPE shortcut button.

The practical difference between interval and ratio MDS is that in interval MDS uncertainty about the true zero point of the D_{ij} data is acceptable. In certain situations this is very important. For instance, when using Likert-like scales it can happen that one is uncertain that test subjects are responding such that the bottom-level response represents a true zero point for the construct under study. Special Case 6 provides data that illustrate how switching from a ratio analysis to an interval analysis can clarify the hidden structure in such a data set. Note that if your D_{ij} values come from some source that has a reliable zero-to-one range, such as correlation coefficient data, then using interval MDS is not appropriate. You should not introduce a new parameter (IC in this case) just to find a lower objective function value. Any new parameter must have sufficient "face validity" that its critics will agree that it makes sense to add it to the model.

Interval MDS has several quirks that may be confusing to the new user. First, all simplex cases (this includes the regular tetrahedron and the 10-object and 24-object simplex examples provided in Permap's library of special cases) have, by definition, all non-diagonal D_{ij} values equal. If you are allowed to shift the D_{ij} zero point, as interval MDS does, a valid solution can be found by reducing all effective D_{ij} values (i.e., $IC + k_1 D_{ij}$) to zero and thus allowing the objective function to go to zero. Thus, interval analyses of simplex data will always result in all objects being superimposed. This is not a degenerate solution. It is a true solution. Second, adjusting the interval constant to minimize the objective function may or may not minimize R^2, the figure of merit used in the Links Evaluation screen. Usually, minimizing the objective function and minimizing R^2 go hand-in-hand, but not necessarily so. It depends on the objective function that is being used. Try using SStress, Interval, and the Three Perpendicular Vectors special case data. The expected "solution," that is, the analog to the Stress, Multiscale, and Fractional cases, will not occur. Finally, there are strong bounds on the value of the interval constant. This is an important point that deserves expansion.

Calculation of the interval constant can be done in two ways. A strict interpretation holds that, to make sense in terms of what a dissimilarity means, an effective dissimilarity ($IC + k_1 D_{ij}$) must be bounded by zero and one. So, if your dissimilarity data set has even one $D_{ij} = 0$ entry then the interval constant cannot be less than zero, and if it has even one $D_{ij} = 1$ entry then the interval constant cannot greater than zero. In these situation, changing from ratio to interval MDS will make no difference at all because the interval constant can have no value other than zero.

5.1.3.4 Interval + Bounds

Interval + bounds analyses are similar to ratio + bounds MDS analyses. Please see Section 5.1.3.2 above. The objective function is not penalized as long as d_{ij} is within the range $IC + k_1 (D_{ij} \pm e_{ij})$ where e_{ij} represents the error or uncertainty in D_{ij}.

5.1.3.5 Ordinal

An ordinal analysis seeks d_{ij} values such that if $D_{ij} < D_{kl}$ then $d_{ij} < d_{kl}$. This means that the objective function is not penalized as long as the calculated d_{ij} are in the same rank order as the D_{ij}.

When an ordinal analysis is specified, Permap treats ties using the so called "primary" convention. That is, all objects of equal rank need not have the same position on the map.

5.1.4 Object Centering

The nonmetric forms of MDS, i.e., ratio + bounds, interval + bounds, and ordinal may only determine the object positions to within certain limits. Because all object positions that are "within the bounds" or "in the right order" contribute zero to the objective function value, there may be infinitely many possible solutions. In reality, these alternative solutions usually are tightly constrained and thus the map does not have any variation at all. However, if the error bounds are large, or if an ordinal analysis is being made with only a few objects, having multiple equally valid solutions can be objectionable. In this case, one might want to ask that the objects be placed at the center point between the error bounds or at the mid point between objects of adjacent ranks.

Centering can be achieved by adding a "junior term" to the objective function. This term must be small enough that it does not affect an object's position when that position is being determined by balancing conflicting dissimilarities, but it must be large enough that it can "nudge the objects over" when they are "between the bounds." This goal is achieved by adding a term to the objective function that is one tenth as large as the normal terms, and is anchored to the applicable midpoint rather than being zero anywhere inside the applicable boundaries. Note that when the centering option is used the value of the objective function increases very slightly. This is a necessary artifact of the centering procedure. It does not indicate that the solution is inferior in any way. It simply means that another requirement has been placed on the solution and that the "cost" of this extra requirement is being reflected in the value of the objective function.

Centering is controlled by the third option box on the right side of the MDS Types and Error Bounds selection screen. Of course, if you are using regular (i.e., no bounds) ratio or interval MDS then the centering feature has no affect.

5.2 Objective Functions

The second submenu of the Analysis Parameters menu allows specification of the objective function, which involves specifying the form of the objective function, the badness function, and the distance measure. The badness and distance problem parameters can be controlled using the Badness and/or Distance shortcut buttons on the main screen. This allows you to quickly scan a variety of solution types while watching the map as it shifts to satisfy the new settings. The Objective Functions submenu can be opened using ctrl-B.

5.2.1 Objective Function Form

The objective function goes by many different names. For instance, some disciplines call it an error function, loss function, criterion function, merit function, and so forth. It is simply a mathematical formula that is maximized or minimized in the process of solving a problem. It is the accumulation of one or more terms that get bigger (or smaller in the case of a merit function) as the objects are moved away from a good configuration.

In some early MDS papers the objective function was completely unspecified. The steps used to optimize "something" were described, but exactly what was being optimized was never defined! This may seem almost unbelievable to new MDS researchers, but it makes sense when you consider the progress that has been made in our ability to quickly make a large number of calculations. Before the early 1960s computational difficulties were so great that the precise details of making a MDS map took precedence over the bigger picture involving the mathematics of the problem's goal and constraints. Nowadays, just the opposite is true. This manual, for instance, reflects the trend of emphasizing the description of the problem, assumptions, and constraints, rather than the solution procedure. This separation can be noted by the little space given to numerical analysis techniques (Section 10.0) compared to the great amount of space given to the definition and description of the problem and solution (Sections 1.0 through 8.0).

Most objective functions are defined to be the sum over all object pairs of the pair's weight factor times the square of the pair's badness measure B_{ij}. This is called a weighted sum-squares form. Mathematically, the sum-squares objective function equation is:

Sum-Squares form: Objective Function = $[\text{SUM}(\text{all } i, j < i)\ W_{ij}\ B_{ij}^2\ ] / NC$

Ignore for now the normalizing constant, NC. It will be described in Section 5.2.4. Focusing only on the form of the above equation, one can see how the individual badness terms are combined to form the quantity to be minimized. This separation between the definition of B_{ij} and how the B_{ij} are accumulated represents a level of rigor that is not always adhered to. Some authors implicitly assume a form for the combination of the badness terms and then use the badness name for the name of the objective function. This causes some confusion. For example, if one says that Stress is the criteria, does that specify how badness is measured or how the entire objective function is defined? Both usages can be found in the literature. Here, we takes Stress, Stress1, SStress, Multiscale, and Fractional to define the badness measure, but it is just as common to find authors that use these terms to specify the overall objective function.

The sum-squares form of the objective function is the most commonly used form. This may be because it is very simple while still being an analytical function (i.e., all derivatives exist) and an even function (i.e., f(x) = f(-x)), or perhaps it is because of the analogy with common statistical estimators that are based on maximum likelihood arguments. For whatever reason, it was used by the originators of the badness terms Stress, SStress, Multiscale, and Fractional. However, you will find other objective functions that use a "root-sum-squares" form. For instance, Kruskal introduced Stress1 and Stress2 (Kruskal's Stress2 never gained wide acceptability and is not discussed further here) that use the root-sum-squares form. Mathematically, the root-sum-squares objective function equation is:

Root-Sum-Squares form: Objective Function = $\text{SQRT}\{[\text{SUM}(\text{all } i, j < i)\ W_{ij}\ B_{ij}^2\ ] / NC\}$

Even though the root-sum-squares form was used originally to define Stress1, you will find some early papers that redefine Stress1 using the sum-squares way of combining the badness terms. Similarly, even though the sum-squares form was used by the originators of SStress, some papers use a root-sum-squares formula for it. Clearly, the situation is confusing. It would be better had the early workers in the field given more thought to maintaining consistency in their definitions, but it didn't happen. The situation is tolerable only because it makes no difference to the configuration of the map whether the sum-squares or root-sum-squares form is used. The only difference lies in the value of the objective function that is reported for a particular map. The root-sum-squares form makes Stress1 values considerably larger than Stress and all the other objective functions which use the sum-squares form. In fact, making the values larger was Kruskal's stated reason for using the square root in his definition.

Permap uses the definition which conforms to the originator's definition. Thus, it is to be understood that the Stress1 badness always uses the root-sum-squares form of the objective function and that all the rest use the sum-squares form. This means that if you use the BADNESS FN shortcut button to cycle through the various badness functions the objective function value will increase significantly when Stress1 is activated whereas the other functions will have values that are of comparable magnitude.

Because MDS maps found using Stress and Stress1 are exactly the same (because the badness function is exactly the same), one might want to eliminate one or the other from the BADNESS FN shortcut rotation list. This can be done using the option box at the lower left side of the Analysis Parameters / Objective Functions screen.

5.2.2 Badness Functions

A badness function is simply a definition of what it is that makes the positions of a pair of objects be bad. Fundamentally, the object pair should be separated by a distance that is consistent with the pair's dissimilarity. Thus, the badness B_{ij} should involve a measure of the mismatch between d_{ij} and D_{ij}. There are several valid ways of defining this mismatch.

Permap offers the five most common types of badness measure, Stress, Stress1, SStress, Multiscale, and Fractional. Because the badness formulas for Stress and Stress1 are identical they are combined here. Remember, however, that Stress and Stress1 imply different methods of combination to form the objective function. Each of the five badness types must be adjusted for the type of MDS being used. We start with Ratio MDS because it most clearly shows the underlying nature of the badness relationship between dissimilarity and distance.

Ratio MDS

Stress & Stress1: $B_{ij} = d_{ij} - D_{ij}$

SStress: $B_{ij} = d_{ij}^2 - D_{ij}^2$

Multiscale: $B_{ij} = \ln(d_{ij} / D_{ij})$

Fractional: $B_{ij} = (d_{ij} - D_{ij}) / D_{ij}$

Ratio + Bounds MDS

Stress & Stress1:

```
If $d_{ij} > D_{ij} + e_{ij}$  Then
        $B_{ij} = d_{ij} - D_{ij} - e_{ij}$
ElseIf $d_{ij} < D_{ij} - e_{ij}$  Then
        $B_{ij} = d_{ij} - D_{ij} + e_{ij}$
Else
        $B_{ij} = 0$
EndIf
```

SStress:

```
If $d_{ij} > D_{ij} + e_{ij}$  Then
        $B_{ij} = d_{ij}^2 - (D_{ij} + e_{ij})^2$
ElseIf $d_{ij} < D_{ij} - e_{ij}$  Then
        $B_{ij} = d_{ij}^2 - (D_{ij} - e_{ij})^2$
Else
        $B_{ij} = 0$
EndIf
```

Multiscale:

```
If $d_{ij} > D_{ij} + e_{ij}$  Then
        $B_{ij} = \ln(d_{ij} / (D_{ij} + e_{ij}))$
ElseIf $d_{ij} < D_{ij} - e_{ij}$  Then
        $B_{ij} = \ln(d_{ij} / (D_{ij} - e_{ij}))$
Else
        $B_{ij} = 0$
EndIf
```

Fractional:

```
If $d_{ij} > D_{ij} + e_{ij}$  Then
        $B_{ij} = (d_{ij} - D_{ij} - e_{ij}) / (D_{ij} + e_{ij})$
ElseIf $d_{ij} < D_{ij} - e_{ij}$  Then
        $B_{ij} = (d_{ij} - D_{ij} + e_{ij}) / (D_{ij} - e_{ij})$
Else
        $B_{ij} = 0$
EndIf
```

Interval MDS

Stress & Stress1: $B_{ij} = d_{ij} - IC - D_{ij}$

SStress: $B_{ij} = d_{ij}^2 - (IC + D_{ij})^2$

Multiscale: $B_{ij} = \ln(d_{ij} / (IC + D_{ij}))$

Fractional: $B_{ij} = (d_{ij} - IC - D_{ij}) / (IC + D_{ij})$

<u>Interval + Bounds MDS</u>

Stress & Stress1:

```
If $d_{ij} > IC + D_{ij} + e_{ij}$ Then
    $B_{ij} = d_{ij} - IC - D_{ij} - e_{ij}$
ElseIf $d_{ij} < IC + D_{ij} - e_{ij}$ Then
    $B_{ij} = d_{ij} - IC - D_{ij} + e_{ij}$
Else
    $B_{ij} = 0$
EndIf
```

SStress:

```
If $d_{ij} > IC + D_{ij} + e_{ij}$ Then
    $B_{ij} = d_{ij}^2 - (IC + D_{ij} + e_{ij})^2$
ElseIf $d_{ij} < IC + D_{ij} - e_{ij}$ Then
    $B_{ij} = d_{ij}^2 - (IC + D_{ij} - e_{ij})^2$
Else
    $B_{ij} = 0$
EndIf
```

Multiscale:

```
If $d_{ij} > IC + D_{ij} + e_{ij}$ Then
    $B_{ij} = \ln(d_{ij} / (IC + D_{ij} + e_{ij}))$
ElseIf $d_{ij} < IC + D_{ij} - e_{ij}$ Then
    $B_{ij} = \ln(d_{ij} / (IC + D_{ij} - e_{ij}))$
Else
    $B_{ij} = 0$
EndIf
```

Fractional:

```
If $d_{ij} > IC + D_{ij} + e_{ij}$ Then
    $B_{ij} = (d_{ij} - IC - D_{ij} - e_{ij}) / (IC + D_{ij} + e_{ij})$
ElseIf $d_{ij} < IC + D_{ij} - e_{ij}$ Then
    $B_{ij} = (d_{ij} - IC - D_{ij} + e_{ij}) / (IC + D_{ij} - e_{ij})$
Else
    $B_{ij} = 0$
EndIf
```

<u>Ordinal MDS</u>

Stress & Stress1:

```
If $Ld_{ij} < Hd_{ij}$ Then
    If $d_{ij} < Ld_{ij}$ Then
        $B_{ij} = d_{ij} - Ld_{ij}$
    ElseIf $d_{ij} > Hd_{ij}$ Then
```

```
                        $B_{ij} = d_{ij} - Hd_{ij}$
                Else
                        $B_{ij} = 0$
                EndIf
        Else
                If $d_{ij} > Ld_{ij}$  Then
                        $B_{ij} = d_{ij} - Hd_{ij}$
                ElseIf $d_{ij} < Hd_{ij}$  Then
                        $B_{ij} = d_{ij} - Ld_{ij}$
                Else
                        $B_{ij} = 2d_{ij} - Ld_{ij} + Hd_{ij}$
                EndIf
        EndIf


SStress:        If $Ld_{ij} < Hd_{ij}$ Then
                If $d_{ij} < Ld_{ij}$  Then
                        $B_{ij} = d_{ij}^2 - Ld_{ij}^2$
                ElseIf $d_{ij} > Hd_{ij}$  Then
                        $B_{ij} = d_{ij}^2 - Hd_{ij}^2$
                Else
                        $B_{ij} = 0$
                EndIf
        Else
                If $d_{ij} > Ld_{ij}$  Then
                        $B_{ij} = d_{ij}^2 - Hd_{ij}^2$
                ElseIf $d_{ij} < Hd_{ij}$  Then
                        $B_{ij} = d_{ij}^2 - Ld_{ij}^2$
                Else
                        $B_{ij} = 2d_{ij} - Ld_{ij}^2 + Hd_{ij}^2$
                EndIf
        EndIf


Multiscale:     If $Ld_{ij} < Hd_{ij}$ Then
                If $d_{ij} < Ld_{ij}$  Then
                        $B_{ij} = \ln(d_{ij} / Ld_{ij})$
                ElseIf $d_{ij} > Hd_{ij}$  Then
                        $B_{ij} = \ln(d_{ij} / Hd_{ij})$
                Else
                        $B_{ij} = 0$
                EndIf
        Else
                If $d_{ij} > Ld_{ij}$  Then
                        $B_{ij} = \ln(d_{ij} / Hd_{ij})$
        ElseIf $d_{ij} < Hd_{ij}$  Then
            $B_{ij} = \ln(d_{ij} / Ld_{ij})$
        Else
            $B_{ij} = \ln(d_{ij} / Hd_{ij}) + \ln(d_{ij} / Hd_{ij})$
        EndIf
    EndIf
```

Fractional:

```
    If $Ld_{ij} < Hd_{ij}$ Then
        If $d_{ij} < Ld_{ij}$  Then
            $B_{ij} = (d_{ij} - Ld_{ij}) / Ld_{ij}$
        ElseIf $d_{ij} > Hd_{ij}$  Then
            $B_{ij} = (d_{ij} - Hd_{ij}) / Hd_{ij}$
        Else
            $B_{ij} = 0$
        EndIf
    Else
        If $d_{ij} > Ld_{ij}$  Then
            $B_{ij} = (d_{ij} - Hd_{ij}) / Hd_{ij}$
        ElseIf $d_{ij} < Hd_{ij}$  Then
            $B_{ij} = (d_{ij} - Ld_{ij}) / Ld_{ij}$
        Else
            $B_{ij} = (d_{ij} -Ld_{ij})/Ld_{ij} +(d_{ij} -Hd_{ij})/Hd_{ij}$
        EndIf
    EndIf
```

The ratio and interval badness definitions are taken from the literature. The ratio + bounds and interval + bounds definitions flow from the ratio and interval formulas after being adjusted for a "zero penalty" zone centered about the target D_{ij}. Ordinal definitions are constructed similarly to the ratio + bounds definitions, except they no longer reference D_{ij}. For ordinal MDS all relationships are relative. The D_{ij} come into play only in determining the rank order of each i, j term. The new variable Ld_{ij} stands for the distance between the two objects that have the next lowest dissimilarity beneath D_{ij}. Similarly, Hd_{ij} stands for the distance between the two objects that have then next highest dissimilarity above D_{ij}.

The ratio MDS Multiscale and Fractional badness measures have singularities at $D_{ij} = 0$, and the Multiscale measure has another singularity at $d_{ij} = 0$. In practice, 0.0001 is added to the vulnerable terms to avoid these singularities. This is a common technique used in numerical analysis. It maintains a monotonic relationship with the unmodified formulas, is computationally efficient, and causes insignificant (invisible under normal viewing conditions) distortions to the map. Technically speaking, the introduction of the 0.0001 terms means that the Multiscale and Fractional measures should be called Pseudo

Multiscale and Pseudo Fractional measures, but this refined terminology is essentially never used in the literature and is not used here.

The choice of the badness measure is not always important. There are numerous comments in the literature that this aspect can be ignored with the expectation that the map will be unchanged. Because of this, some researchers simply assume that the Stress badness formula is the proper choice. However, the conclusion that these details are not important is not always true. A little work with Permap will show that it probably results from a false generalization based on data sets that are well fit. The better the fit the more the choice of the badness measure is immaterial. However, if the fit is not good, then the choice can be very important. Once you have Permap up and running, check out the differences in configuration for the Stacked Triangles test case when SStress is used instead of a Stress. It is important that the affect of this basic problem parameter be tested for each different data set. You will find that the more stressed your solution is the more important the choice is.

Finally, it should be mentioned that the Multiscale measure really is of very little value. When used with ordinal MDS, it is of even less value. It is included because, to the best of our knowledge, the typical MDS researcher likes to "try everything" and "see for themselves." So it is included more for satisfying curiosity than for use. Here is why Multiscale is not very useful. First, it does not conform to most people's intuitive definition of badness. That is, it is highly non-linear in its spatial relationship between badness and displacement. Second, it leads to unstable mappings. In fact, unless your data set has the possibility of a low-stress mapping (such as many of the special cases data sets do) using Multiscale may produce wildly unstable results. Of course, you can first solve the problem using Stress or SStress and then rotate to Multiscale to see what the Multiscale mapping looks like, but direct convergence from random starting points is difficult to achieve. Even using a smaller step size doesn't help in most cases. So, if you are a "new user in a hurry," then just ignore Multiscale. Otherwise, enjoy, but don't get upset if your data set doesn't have a stable solution.

5.2.3 Distance Measures

To determine the badness, B_{ij}, you have to have the distance between two objects, d_{ij}. To measure d_{ij} you must have a definition of how distances are measured. For example, if you have two points (X_1, Y_1) and (X_2, Y_2) on a map, how far apart are they? If you believe in the Euclidean measure, then the distance is the length of a straight line between the points. If you believe in the City Block measure, then the distance is the absolute value of X_1-X_2 plus the absolute value of Y_1-Y_2, which is equivalent to taking an "up and over" path. If you believe in the Dominance measure where only the larger of the two differences is considered to be important, then the distance is the maximum of the absolute value of X_1-X_2 and absolute value of Y_1-Y_2. These and other metrics can be represented using the general Minkowski formula where the exponent p has a value that can vary from one to infinity.

Euclidean: $d_{ij} = SQRT\{(X_i-X_j)^2 + (Y_i-Y_j)^2\}$

City Block: $d_{ij} = |X_i - X_j| + |Y_i - Y_j|$

Dominance: $d_{ij} = MAX(|X_i - X_j|, |Y_i - Y_j|)$

Minkowski: $d_{ij} = \{|X_i - X_j|^p + |Y_i - Y_j|^p\}^{1/p}$

Further generalization of the Minkowski form involves applying weights to the various X_i-X_j and Y_i-Y_j differences (Hartigan, 1975, p. 64). However, these extensions are essentially never seen in the applied MDS literature. In fact, the general Minkowski form and the Dominance form (equivalent to Minkowski $p = \infty$ but often approximated by using $p = 30$) is almost never seen. In practice, only the City Block (equivalent to Minkowski $p = 1$) or Euclidean (equivalent to Minkowski $p = 2$) distance metric is used to any significant extent. Between City Block and Euclidean, the Euclidean is overwhelmingly the most commonly used metric. Using the City Block metric is recommended only if you are knowledgeable about the perceptual assumptions that go with it. See Torgerson's early work (Torgerson, 1952, 1958, 1965) for a detailed discussion of the relationship between spatial and psychological distances, and see Hubert, Arabie and Hesson-Mcinnis (1992) for a discussion of the mathematics of the City Block measure. One could go so far as to say that you should always use the Euclidean metric unless you are thoroughly grounded in the fundamentals of measuring differences between abstract constructs. Borg and Groenen (1997, p. 14) emphasized this position when they wrote ". . . MDS representations that employ distance functions other than the Euclidean tend to be misleading when inspected intuitively. Therefore, they are useless for exploratory purposes." Despite the possibility of misuse, Permap includes the City Block and the general Minkowski distance measures to let users explore for themselves the various distance measurement alternatives.

A final detail about the Minkowski distance measure needs explanation. The Minkowski formula needs double precision arithmetic in order for the calculated distances to have the number of significant figures needed for Permap to be able to calculate the value of the objective function to many significant figures. Double precision arithmetic is slower than single precision arithmetic and is not used in Permap. Thus, if you use a Minkowski distance formula you may find that for some p values the objective function is not as stable as it would be if a different distance function was used.

5.2.4 Normalization Constants

Normalization of a quantity is achieved by dividing it by a constant, NC that is chosen to achieve three purposes. First, you want the units of measurement to cancel out. That is, you would like to get the same results regardless of whether the original data were expressed in centimeters or inches. Second, you want large problems to produce the same intensive results as small problems. This means the NC must serve as a scale factor. Third, you want values to fall in the zero-one range so that the results based on them will be easily interpreted and do not require scale-related weights. This last goal is the overarching goal of normalization, but it cannot always be achieved such as when the quantity being normalized is not bounded. Fortunately, this is not a problem for MDS objective functions.

Normalization of the objective function does not influence the resulting MDS map at all. It only changes the numerical value of the objective function value. Still, to facilitate comparisons across studies it is best to normalize the objective function. Therefore, Permap always displays normalized objective function values.

Permap uses the following NC factors:

Stress:	NC = Sum (all i, j < i) W_{ij} D_{ij}^2
Stress1:	NC = Sum (all i, j < i) W_{ij} d_{ij}^2
SStress:	NC = Sum (all i, j < i) W_{ij} D_{ij}^4
Multiscale:	NC = Sum (all i, j < i) W_{ij}
Fractional:	NC = Sum (all i, j < i) W_{ij}

By convention, if W_{ij} is not used then they are set equal to one. In this case the sum over the lower left corner of the weights matrix reduces to N (N - 1) / 2 where N is the number of objects in the analysis.

Finally, if you carefully examine the preceding formulas you notice that there is an inconsistency in the Stress1 NC definition. The normalizing constant for Stress1 is not a constant at all. So, in this case, the term "normalizing constant" should be replaced with "normalizing factor." The imprecision is tolerated to gain the pedagogical value of the taxonomy and to avoid arguments with Stress1 advocates.

5.3 Mapping Weights

Before discussing mapping weights, it is helpful to briefly digress into weight factors in general. Some MDS analyses involve assigned weights W_{ij} that multiply the corresponding B_{ij}^2 terms. Permap allows you to enter these weights through the data file. They reflect the importance, uncertainty, saliency, flexibility, etc., associated with each object-to-object dissimilarity value. Sometimes they are taken to be the reciprocals of the variances in the experimental data and sometimes they are just subjective assignments. Most commonly they are not used. If Permap finds no weight information in the data file, then all W_{ij} are set equal to one.

Note that if the dissimilarities are well fit by the map (the minimum objective function value is zero) then the weights are immaterial. The map will remain stationary for any random assignment of weights because the weights are multiplying zero badness quantities. However, if the fit is not perfect, then the weights tell how to respond to each pair's badness. You might say they provide a method for determining "what yields the most." For example, if you are unsure of one object pair's dissimilarity, and it conflicts with those of one or more other pairs that you know with some certainty, then you should give the more uncertain B_{ij} term a lower weight.

Now let's shift our thinking away from traditional weights to a new kind of weight factor called a mapping weight. Mapping weights are denoted MW_{ij}. They are built into Permap and controlled by the Analysis Parameters/Mapping Weights menu or the Mapping Weights shortcut button. These mapping weights do not replace the regular weights W_{ij}. They multiply the W_{ij} weights. It is the product W_{ij} MW_{ij} that is used in the calculations. Thus, if you decide to use mapping weights, then everywhere W_{ij} appears in this manual you should replace W_{ij} with W_{ij} MW_{ij}. To emphasize this point, we make it a separate subsection.

5.3.1 Replacing W_{ij} with W_{ij} MW_{ij}

If you choose to use mapping weights, then all formulas involving W_{ij} should replace W_{ij} with W_{ij} MW_{ij}.

5.3 Mapping Weights, continued

The importance and general applicability of mapping weights is not widely recognized. There are three distinct reasons why mapping weights might be appropriate.

1. Ignoring questions of scale and orientation the statement that zero dissimilarity exists is meaningful whereas the statement that complete dissimilarity exists may or may not even be capable of precise definition.

2. Percent uncertainty values, rather than absolute uncertainty values, often better conform to measurement reality.

3. A simple Cartesian map will not always convey the essence of the underlying relationships.

The first of these points is fundamental. It involves the issue of knowability. Let (i, j) be (3, 4) to simplify the discussion. The information content of saying $D_{34} = 0$ is quite different from saying $D_{34} = 1$. Even in the simple case where two objects can be described by specifying the value of its attributes the problem remains. Saying that $D_{34} = 0$ means that all corresponding attribute values of object 3 and object 4 are identical. However, the same type of statement is not valid for the complete dissimilarity case. If you know the nature of object 3 and object 4 then saying that $D_{34} = 1$ does not mean that you know the nature of object 4. To put it still another way, test subjects might be able to explain why they consider two objects to be identical, but they cannot explain in similar detail why they consider two other objects to be completely different. This means that position misfits between objects with a small dissimilarity should not be treated like position misfits for objects with large dissimilarities. Strongly dissimilar values should be given less weight in determining the shape of the map. In other words, mapping weights should decrease with increasing dissimilarity values. The importance of this point is emphasized when one realizes that most objective functions do just the opposite. They give more weight to large D_{ij} values than small D_{ij} values because the function involves the squaring of D_{ij}.

The second point is practical. It is often true that the degree of uncertainty in a measurement is roughly proportional to the magnitude of the property being measured. This is not a necessary relationship, but one that is frequently observed. If it applies, then instead of spending considerable time estimating W_{ij} for each dissimilarity, it is easier to leave all W_{ij} with the default setting $W_{ij} = 1$, and let Permap generate appropriate mapping weights that decrease with increasing dissimilarity values.

The third point is fundamentally different from the first two. It does not involve using any concept of uncertainty. Instead, it involves recognition that a "photograph like" picture of reality is not always the best picture to explain the essence of the matter. This is why Picasso would take the elements of a traditional picture and move them about, change their orientation, change their color, or distort them in some manner. In doing so he would facilitate a new understanding of the subject. Mapping weights can do this. Consider a set of eight objects that are arranged in three-dimensional space as if they were at the corners of a long box-kite (Permap's Special Case 7). Now use the Stress badness measure, the Euclidean distance measure, two dimensions, and make a ratio MDS map of the eight objects. The resulting map shows the outline of a long rectangle made up of four symbols in an almost straight line, parallel to and remote from another four symbols in an almost straight line. This is a valid two-dimensional picture of the three-dimensional box-kite. It faithfully shows the two remote subgroups and some of the symmetry involved. However, it badly distorts the relationships existing within the two subgroups. If mapping weights are applied (click the MAPPING WTs. shortcut button twice), the MDS map immediately shifts to two squares remote from each other. Which picture is correct? Both are, just as Picasso's art is as valid as Rembrandt's.

Another example of how the use of mapping weights can clarify hidden structure in a data set is given in EXAMPLE_C.txt. Here, Churchill defines ten department store characteristics to be the objects of interest. He determines the correlations between objects based on a set of shoppers' impressions of different department stores. After loading Permap use Stress badness measure, the Euclidean distance measure, two dimensions, and make a ratio MDS map of the data in EXAMPLE_C.txt. You will find, as Churchill did, that there are two groups of characteristics. One group has six members and the other four members. Next, use the MAPPING WTs. shortcut button to rotate to using linear mapping weights. The group of six is only slightly changed, while the group of four takes on a significantly altered internal arrangement. Which picture is correct? As before, both are correct. It is up to the researcher to determine if the new configuration contributes to understanding the topic.

Permap calculates mapping weights internally based on the values of the given D_{ij}. File entry of MW_{ij} values is not supported. The built-in mapping weight function is:

$$MW_{ij} = (1 - D_{ij}) / (1 - m\ D_{ij})$$

where "m" is an adjustable parameter with a value less than or equal to one.

Experience shows that most interesting behavior can be investigated using m = 1 (no mapping weights, equivalent to $MW_{ij} = 1$), m = 0.95 (weak mapping weights), m = 0 (medium mapping weights, equivalent to a linear decrease of MW_{ij} with increases in D_{ij}), and m = -20 (strong mapping weights). Permap allows selection of these four standard mapping weight functions by using the MAPPING WTs. shortcut button. Or, any arbitrary value of m can be selected using the Analysis Parameters/Mapping Weights menu. This menu also shows a graph of MW_{ij} vs. D_{ij} and displays a table of the corresponding MW_{ij} values.

Note that the above mapping weights function goes to zero at $D_{ij} = 1$ for all m not equal to 1. This would allow independent object groups to "float away" from each other in a numerical analysis sense. To avoid this a "floor" is introduced. The MW_{ij} floor is 0.1 for weak, 0.01 for medium, and 0.001 for strong mapping weights. The variable case (see the Mapping Weights screen) uses a floor equal to MW_{ij} calculated using the above equation with $D_{ij} = 0.98$.

Finding alternate views of reality, as was demonstrated by the box kite example or the department store comparison data of Churchill in EXAMPLE_C.txt, is one use of mapping weights. Another is to check for conflicts that might exist between large and small dissimilarity values. To demonstrate this point, load data file EXAMPLE_A.txt which contains the Ekman (1954) color comparison data. Find the two dimensional map using ratio MDS, the Euclidean distance metric, and the Stress badness function. You will get the standard circular figure that is shown in many text books. Then, click the MAPPING WTs. shortcut button (use the View menu to turn on this feature if it does not already appear on the main screen) twice to reach the medium mapping weights" selection. This will reduce the influence (weight) of the large dissimilarity values vis-a-vis the small ones. The map will shift significantly, thus showing that there is a systematic conflict between the positions of the objects with large dissimilarities versus the positions of the objects with small dissimilarities. Further analysis will show that the large dissimilarities favor the ring shape and the small dissimilarities act to disrupt the symmetry of the ring. If this was your research, you would next decide which picture is most appropriate for the goals of your research. Or, perhaps you would decide that further experimentation is called for in order to refine your estimates of the dissimilarities.

While considering the Ekman results, it is worthwhile to note the danger of reading too much into circular patterns. The Ekman circle may be as significant as several articles claim, but it could be that it resulted simply because it is easier to quantify differences between similar colors than it is to quantify differences between dissimilar colors. If such were the case then colored objects would have their near-neighbor relationships well represented while the remote relationships would not be well represented. This form of non-linearity can result in meandering chain-like patterns being turned into circular MDS patterns. For example, consider this exaggerated case. A closed chain of objects has all of its nearest-neighbor dissimilarities coded as zero and all of its non-nearest-neighbor dissimilarities coded as one. Then, as is shown by analyzing the three different data sets in EXAMPLE_D.txt, a two-dimensional perfect circle is found. The circle is a valid MDS solution, but it suggests much that is not necessarily true. As the number of objects increases from three to, say, six or seven objects, the danger is even greater. In these cases, a simplex matrix (i.e., a matrix with all non-diagonal dissimilarity values equal) always yields a circular symmetry. This means that if your respondents are saying that the less similar objects are "all about equally different" you will discover a circular relationship in your MDS map. Thus, one must be careful not to draw quick conclusions after finding objects in a circular pattern.

5.4 Dimensionality and Boundaries

MDS analyses are not fully described until two more aspects are covered. You must specify how many dimensions are to be used to describe the object placements, and what boundary conditions (external constraints) are to be placed on the solution. These aspects can be specified using the Analysis Parameters/Dimensionality and Boundaries submenu (ctrl-Y).

5.4.1 Dimensionality

Pressing the Dimensions shortcut button or using ctrl-Y to open the Dimensionality and Boundaries screen allows you to control the dimensionality of your MDS solution. Permap can analyze a problem using one to eight dimensions. That is, the solution can be described in terms of each object's coordinates in any of eight solution spaces. Permap can write out the object's coordinates to the PermapSolutionLongFmt.txt file if you want to see the actual values. But, the important question is, since Permap can display only one or two dimensional maps, why would you want it to make a three or four dimensional analysis? The answer to this question may not be what you expect.

Let's take the easier of the two cases first. Why would you want to make a four dimensional analysis? Given that four dimensional configurations cannot be visualized, what good are they? MDS is inherently a visual tool. Providing a way to transform a set of numbers into a picture is its reason for being. But in this case the visual nature of the solution is not of interest. The answer is, we make the four dimensional solution not to understand the data's hidden structure in four dimensions but to check on the validity of a three dimensional solution. There are many ways to test the merit of a particular solution. One can use the objective function value, a Shepard plot, a Pareto plot of objective function components, Waern link plots, attribute gradient plots, and so forth. Each of these is useful and should be part of a complete analysis. But the most direct way to see if a three dimensional map is stable, and thus "good", is to remove the restriction to three dimensions and see if any of the objects move. This is called "relaxing the solution into a higher dimension." If a four dimensional solution gives the same picture as a three dimensional solution, then you have strong evidence that three dimensions are adequate for describing your data set.

Answering the similar question of why you would want to make a three dimensional analysis follows the same logic as that for four dimensions. The only difference is that, as opposed to four dimensional results, three dimensional results can be visualized. So why does the above logic still apply? It applies because we need to communicate our results to others and many times these other people, especially those with the financial, organizational, or political clout to do something with the information, cannot understand the meaning of a collection of points in three dimensional space. So, in many cases, the value of making a three dimensional analysis lies more in its ability to help evaluate the quality of a two dimensional map rather than in the utility of the three dimensional solution itself.

In light of the preceding two paragraphs, why would one ever want to solve problems in more than four dimensions? An adequate reason is that it is fun to "look around" in higher dimensions. However, a more common reason is to make a skree analysis that is described in many MDS texts. A skree analysis involves solving a problem using many different dimensions, then plotting each solution's Stress versus the dimensionality of the solution space, and then taking the problem's optimal number of dimensions, D^*, to be the number of dimensions used at the plot's "elbow." The plot is supposed to look like the skree (loose rock) and valley floor at the base of a mountain. It is supposed to distinctly show D^* where the two straight lines ("the skree line and the valley floor line") meet. The trouble with this very reasonable approach is that it doesn't work. It can be made to appear to work by illustrating the analysis using a hypothetical data set that is well fit using D^* dimensions. The tip-off that artificial data are being used is noticing that Stress is very small when using D^* dimensions. Unfortunately, real data sets don't behave this way. From experience one can say that the Stress-Dimensions curve will have a gradual bend, not a sharp elbow, and the analyst can pick D^* to be pretty much whatever number of dimensions he or she prefers. Admittedly, that is a bit too harsh of a description of what happens with real skree analyses. To compromise, the authors will happily soften their conclusion once they are sent a real, unmanipulated and unprocessed data set that shows a clear break in the optimal solution's Stress vs. dimensionality plot, AND the Stress using D^* dimensions is not nearly zero.

Finding a good X-Y plot for a three or more dimensional solution is not always easy. It is important to understand the difference between starting an analysis in, say, three dimensions versus starting it in two and then relaxing to three. Because Permap displays a two dimensional map, when three or more dimensions are being used it is the projection on the X-Y plane that is displayed. (Caution, this does not mean that a two dimensional solution is simply a projection from a higher dimension. This is a point of confusion for some new users.) Whenever you start a new solution Permap assigns random positions to the coordinate values of each object and then makes the analysis. So, each different "run" produces a different "view." That is, the X-Y projection shown is different even though the relative object positions are the same. These changing views can be confusing at first, but they are very valuable because they give a much richer understanding of the three dimensional shape of the object cluster.

If you prefer to always have the X-Y plane hold the maximum amount of variation, and usually you will, then you can do two things. One, start the analysis in two dimensions and subsequently relax it into three dimensions. Two, instruct Permap to try to adjust the view for you. When making three or more dimensional analyses, Permap's algorithm can be instructed to favor X-Y maps that have large cross-sections. Permap is not always successful in this endeavor. Exercising this option is done by checking the box "Attempt to keep the maximum ..." at the bottom right of the Dimensionality explanation area. Beginners are advised to check the box.

To better understand shifting views of a solution, load Special Case 3 which has a dissimilarity matrix based on three connected perpendicular vectors in three dimensional space. That is, the data represent the normal Euclidean axes near the origin. Use a ratio analysis, Euclidean distance measurement, and check the "Attempt to keep the maximum ..." dimensionality option. Solve the problem in three dimensions. The map will most probably show what you expect. That is, a figure with three-part circular symmetry. Now hit Start numerous times. The figure will not change significantly, except for irrelevant rotations and perhaps a few random local minima solutions. Permap does a good job of keeping the maximum cross-section in the X-Y plane. However, if you look very closely you may see some slight changes that are due to Permap's imperfect orientation mechanism. Now uncheck the "Attempt to keep the maximum ..." option and hit Start several more times. It will become immediately obvious that you are seeing the three vectors in 3D space from widely differing perspectives. Sometimes you will be looking down on one of the three vectors and it will effectively disappear. Sometimes one or two of the vectors will be greatly foreshortened. Practice with this special case will develop your ability to understand the role of viewpoint in understanding three dimensional solutions.

Turning now to MDS analyses in one dimension, it should be noted that these analyses are particularly difficult. Local minima abound. This is because the objects cannot "go around" each other when seeking their best positions. If the objects cannot go around each other then they would have to "go over" or "go through" each other to reach their best positions, and that can make for difficult mathematics. There are two ways to lessen this difficulty. First, the SStress badness function is more effective than any of the other badness functions when it comes to letting objects get past blocks. If you need to use one of the other badness functions, then first use SStress and after its solution is found change to your preferred badness function. Second, make the analysis in two dimensions, then use the field commands to rotate the map's longest span to the horizontal, and then change back to one dimension and continue the solution. Neither of these techniques will always work, but they are worth trying.

5.4.2 Boundaries

Pressing the Boundary shortcut button, or using ctrl-Y to open the Dimensionality and Boundaries screen, allows you to apply boundary conditions (external constraints) to your MDS solution. It is surprising that few MDS programs provide for placing even simple boundary conditions on MDS maps despite the fact that the topic is well developed in the literature. Lee and Bentler (1980) were one of the first that discussed the subject and Borg and Groenen (1997, p. 182) provide a summary of work on the topic. Applying boundary conditions, or constraints as they are sometimes called, is very common in the work of physical scientists, but behavioral scientists have not placed much emphasis on it. While it is true that many "conceptual space" MDS analyses do not have obvious boundary conditions, many others do and the possibility of their existence should be considered in every MDS analysis.

The idea of boundary conditions is a simple one. For example, if you know all of the person-to-person affinities of a group of people you can solve for their preferred positions relative to one another. However, the solution implicitly assumes that there is space available for the preferred positions and this may not be the case. Perhaps the people are in a small room or maybe they must congregate in a room containing tables that get in the way. Numerous similar examples are easy to describe when the objects represent mobile physical entities. A completely different kind of boundary condition example involves time. If you want to superimpose a time axis on a map, you might want to constrain the map's surface to a "path" or ribbon-like configuration. That is, you want to make a "what if" analysis where MDS gives the best possible configuration for a set of dissimilarities and a time-flow hypothesis. You do not want to make a one dimensional analysis because, you believe, it is possible to have multiple threads running side-by-side so a path or multi-lane road is a more appropriate conceptual model. Maybe you are studying certain hominid artifacts and you know that more than one hominid species developed during the time period of interest. By constraining the map to a long, narrow area you can quickly see what happens when you ask for a map that is compatible with a time-flow hypothesis. A final example involves boundary conditions based on theoretical arguments. It could be that, say, only four different object groupings are theoretically possible. In this case you would want to have the MDS solution fit on a map that allows objects in at most four distinctly different areas.

Permap allows studying the effect of three kinds of boundaries. First, it allows imposing a bounding circle on the group of objects. In essence, the objects are embedded in a space where you can control the size of the space. If a three dimensional analysis is used then the boundary circle becomes a boundary sphere, and if a four dimensional analysis is used then it becomes a hyper-sphere. This is necessary because if the nature of the boundary did not change with the solution's dimensionality then objects in a higher dimensionality could just move "off the plane" and completely avoid the constraint. Second, Permap allows restricting the objects to a narrow path. The path becomes a tube in three dimensions and a hyper-tube in four dimensions. Third, Permap allows restricting the objects to a square area. The square becomes a cube in three dimensions and a hyper-cube in four dimensions.

When applying a boundary one might think that all solutions will simply consist of the objects being pushed up against the boundary. This is not always the case. Some experimentation with the special cases built into Permap will reveal interesting behavior. The implication for the MDS analyst is that when dealing with mobile objects it is important to consider the possibility that constraints are limiting the (X, Y) positions. This means that when constructing a MDS map sometimes it is important to be able to apply boundary conditions that make the results fit both the D_{ij} data and the environment that surrounds the objects.

This discussion of boundaries might prompt you to ask “Do the constraints imposed on certain (X, Y) values have to come from some physical surrounding barrier?” The answer is “no.” For instance, legal rulings, family or societal mores, religious beliefs, or a host of other “forces” may dictate that certain members of a group be, for instance, no closer than some given distance. Note that in all of these cases one or more (X_i, Y_i) positions are being influenced by an external force, not something that involves changing any D_{ij} values. Such restrictions can be viewed as forms of “boundary conditions,” albeit not physical boundaries. This topic is touched on in the section on “fixed locations” although in that section the language of boundaries is not used.

5.5 D_{ij} Precision Degradation

Before discussing data precision, please note that if you are using an attribute matrix to generate your dissimilarity matrix you should read Section 6.1 on attribute data level degradation. This is because if you have attributes then they are the independent variables and you should do "cleaning," "segregating" and “data precision evaluation” work using the independent variables, not the dependent variables (dissimilarities in this particular case).

The topic of precision degradation (a subtopic of the much broader “data discretization” topic) is important. This is because it is important to know if your D_{ij} data's overall precision is adequate to support your results. In other words, every MDS analysis should include a sensitivity analysis.

Before we get into the details of D_{ij} precision degradation, we must answer the question, why are we addressing this subject again? Aren't the error bounds of Section 5.1 sufficient to handle the matter? The answer is that, yes, using those techniques can do the job from a certain perspective. But, the feedback we have received from many Permap users clearly shows that these approaches are not being used as much as they should be. Also, MDS research articles appear every month that contain no sensitivity analyses at all. This is an important flaw. Because we believe making a sensitivity analysis is so important we have changed Permap to make it easier. Starting with version 10.0, an on-line button has been added to Permap's main screen to allow one-click degradation of the D_{ij} data. This button is labeled PRECISION and can be activated with a mouse click or by pressing the "E" key. (If the PRECISION button is not present on the main screen then use the View menu to activate it.) With just a few clicks you can degrade the D_{ij} data's precision all the way back to its (0, 1) roots. If you do this while watching your map, you can immediately draw conclusions about the sensitivity of your map to the precision of your D_{ij} data.

In order to be clear about D_{ij} degradation, we must first review some definitions. "Precision" refers to the degree of refinement with which a measurement is reported. Precision can be characterized in several ways. One is to observe the number of significant figures that are reported, another is to give an uncertainty measure, another is to use the implied rule of ± 5 in the next (missing) decimal place to the right, and so forth. "Accuracy" refers to the degree of conformity between a measurement and a true value. The true value may be known or unknown. That is immaterial. The concept of accuracy defined in terms of a true value is still useful. Finally, the first digit of an actual measurement that differs from a true measurement marks the start of the "false figures." Anytime the precision of a measurement exceeds its accuracy there are false figures involved. We may not know which figures are false, but we can take steps to make sure that they do not invalidate our results. This is particularly important because in various of the “soft sciences” MDS is frequently applied to D_{ij} data of low accuracy and high precision. Clearly false figures are often, some would say essentially always, present.

How does one decide if a MDS map is influenced by false figures? One degrades the D_{ij} data until the false figures disappear. In effect, the increasing error bounds envelope the false figures and they melt away into irrelevancy. Eventually, the error bounds become so large that all critics agree that the true D_{ij} values fall within the bounds. If the map has remained unchanged during the degrading, then it must not be a result of the false figures.

But, if one does not know the true values for the D_{ij} data then one must not know which figures are false, so how can one know when to stop degrading the data? The answer is to degrade the precision "beyond reason," or to degrade it "with your critics at your side." Fortunately, MDS is a very robust procedure. That is, it produces results that are not sensitive to small errors in its input data. Because of this, you will probably find that your MDS map is little changed even after severe, "beyond reason," degradation of its D_{ij} data.

Here is how Permap degrades the D_{ij} data. Beneath the caption "PRECISION" on the precision control button there is an entry that shows the current precision of the D_{ij} data. Starting with the maximum number of decimal places (or maximum number of digits if the input is not already normalized to the zero/one range) in the input D_{ij} data, Permap rounds down to the next lesser number of decimals with each button click. After reaching the tenths-level it starts rounding to the nearest 1/5, 1/4, 1/3, ½, and 1 intervals. At a final interval of 1, all D_{ij} values equal 0 or 1. Another click on PRECISION will return the precision level back to the maximum precision consistent with the input data. It has "Max" appended to the numerical value as a reminder that the input (maximum possible) value is in use.

For example, if $D_{28} = 0.34567$, after one click on the PRECISION button D_{28} will be 0.3457, then 0.346, then 0.35, and then 0.3. Once it reaches the tenths level, Permap begins to degrade the precision using fractional step sizes rather than decimal step sizes. That is, it reassigns the current value to the nearest number on a scale using step sizes of 1/5, 1/4, 1/3, ½, and then 1. This means that, continuing with the example above, D_{28} will change from 0.3 to 0.2, then 0.25, then 0.33, then 0.5, and then to 0. In each case the step size between possible values increases. For example, starting from input data showing five decimals, the measurement interval (step size) varies as follows: 0.00001, 0.0001, 0.001, 0.01, 0.1, 0.2, 0.25, 0.33, 0.50, and 1.0. In each case the uncertainty, i.e. ± half the measurement interval, increases in an analogous manner. As an aside, one might ask why the 1/3 interval is taken to be 0.33 instead of, say, 0.333. The reason is that a general rule from the theory of error analysis is that the last significant figure in any stated answer should be of the same order of magnitude as the uncertainty (Taylor, 1997, p. 15).

The reason for having Permap degrade the precision using fractional intervals after the decimal places are exhausted is that these step sizes, while extreme, correspond to experimental conditions of interest to some researchers. For instance, if a person is asked to assign a dissimilarity to two objects by placing a mark on a line segment that goes from 0 to 1, the dissimilarity value may be reported to the nearest 0.1 value (depending on the length of the line segment, the width of the marker point, and the diligence of the experimenter). Our researcher might be quite happy with the 0.1 precision data. But another researcher might claim that some theoretical principle says that no more than 6 detectable states are possible under the given conditions. This second researcher would insist on recording the dissimilarity values as 0.0, 0.2, 0.4, 0.6, 0.8, or 1.0. So, the first researcher might like to know if his or her MDS map will remain stable if the dissimilarity precision is degraded from the 0.1 interval level to the 1/5 level (five intervals, six possible values, 0.2 interval size). Permap's Precision Button can help answer the question.

Finally, a note of caution is in order for certain degraded precision levels. Sometimes you will notice a significant change in a map at a precision level of ½. The ½ level can cause changes in the map that are even bigger than those occurring when using a measurement interval of 1. This occurs when the D_{ij} data are clustered in the center of the 0 to 1 interval. Under these circumstances the 0.5 ± 0.25 range captures most of the dissimilarities and the D_{ij} matrix moves toward a $D_{ij} = 0.5$ simplex. This behavior is simply an artifact of a poorly conditioned D_{ij} matrix.

6.0 ATTRIBUTES-TO-DISSIMILARITIES

If no attribute information (an A_{ik} matrix) is provided in the data file, then this menu item is disabled (grayed out). If attribute information is provided, Permap uses the attributes to produce dissimilarities and then uses the dissimilarities to produce a map. In this case each object can have up to 100 attributes.

6.1 Attribute Data Types

The first item on the left side of the screen is the Attribute Input Data Level box. Use it to specify the type (i.e., quality or information level) of your attribute data. Acceptable data types are ratio (metric), interval (metric), ordinal (nonmetric), nominal (nonmetric), binary nominal (nonmetric), and "unknown" (both metric and nonmetric). The ordinal level is also known as rank order, and the binary nominal level is also called dichotomous.

Ideally, all of your various different attributes should be of the same data type. For instance, if some attribute values come from Likert data, then all of your attributes should come from Likert data or some other ordinal data of comparable precision. This is the ideal case. In practice this rule is frequently violated. For instance, if you are using the Pearson correlation coefficient it is flexible enough that you can feed it a mixture of ratio, interval, and binary nominal level attribute data. Using binary data with the Pearson formula is not theoretically correct, but the formula works just fine and Permap happily complies with bad instructions of this kind. Other association, correlation, distance, interaction, etc. coefficients are more specific and cannot be used except with one type of data. Permap only goes part way toward solving the problem of mixed data types. It facilitates using mixed types by coercing all the data types into the type that is compatible with the selected attribute-to-dissimilarity function. These coercions are described in Section 6.1.1. This approach is acceptable sometimes and other times it is clearly unacceptable. If this approach is not acceptable and if you have a mixture of data levels in your attributes, then it is up to you to apply an appropriate attribute-to-dissimilarity function to each different type of attribute, and then combine those single-case dissimilarity values into an overall dissimilarity value. See Kaufman and Rousseeuw (1990, p. 32) for a summary of methods for combining diverse attribute information into one dissimilarity value.

When you choose an attribute data type Permap excludes using any attribute-to-dissimilarity functions that do not fit that data type. For example, if you choose to use interval level data then the COS(alpha) = Angular Separation function is prohibited because the COS(alpha) function requires ratio level data. This exclusion is implemented by disabling (graying out) all options in the Attribute-to-Dissimilarity Functions box that are not designed to work with the specified data level, and by causing the Attributes shortcut button on the main screen to cycle through only the allowed functions. As another example, if you choose the ratio level option to describe your input data, then the ratio and interval level attribute-to-dissimilarity functions are activated, the other functions are grayed out, and the Attributes shortcut button is caused to cycle through only the appropriate interval and ratio level functions. The same kind of behavior applies to the other data level selections.

If you choose the "Unknown" data level option, then all attribute-to-dissimilarity functions are allowed. Of course, you should always know the nature of your data. However, the Unknown option has some useful aspects. First, because most MDS users are inveterate experimenters they like to see what happens when the rules are violated. This option is great for doing a little "looking around." Second, as will be explained later, sometimes an expert MDS user will have a valid reason to downgrade the attribute data to be able to use a lower level attribute-to-dissimilarity function. Or, perhaps this would be done to force one’s attribute data to be of one consistent level. After describing how Permap coerces data into different data types we will discuss further why an experienced MDS researcher might want to use the Unknown option and downgrade the attribute information.

6.1.1 Changing Data Types

If you select an attribute-to-dissimilarity function that does not agree with the data type of one or more of your attributes, Permap converts (coerces) the mismatched A_{ik} data into the required format.

Up-Conversions

Up-conversion is somewhat controversial, often interesting, sometimes provides hints about how to improve future data collection, and equally often yields nonsense.

Up-conversions are those conversions that treat lower level data as if they were higher level data. Sometimes this involves nothing more that ignoring reality and sometimes it involves valid manipulations of the data. For example, using interval level data in a ratio level function such as COS(alpha) involves an up-conversion of the first kind. It usually produces nonsense. On the other hand, adding a constant to interval level data in a manner that shifts it to ratio level data (think of converting temperatures on the interval Celsius scale to the ratio Kelvin scale by adding 273.15) involves an up-conversion of the second kind.

Here is how Permap makes up-conversions. When moving from binary nominal level A_{ik} data to a higher level of data Permap just pretends that the binary nominal level data are ratio, interval, ordinal, or nominal and proceeds to use the formulas given in Section 6.2 to calculate the dissimilarities. That is, either it treats the zeros and ones of the binary data as if they were the results of measurements that yielded zeros and ones (for ratio, interval, and ordinal formulas), or it treats them as the names "zero" and "one" (for nominal formulas).

Up-conversion from nominal level A_{ik} data to metric data involves the esoteric matter of ordered lists and unordered lists, and how ordered lists differ from ordinal information (answer, only in their bound set of functions). Because Permap uses numbers as names for nominal data, Permap makes these up-conversions automatically. Equally automatically, the results are usually nonsense.

Up-converting ordinal or interval level A_{ik} data to ratio data is as easy as upgrading binary nominal data. Permap just pretends that the numbers are more than what they really are. If a formula is used that requires ratio level data but lower level data is entered, nonsense often results. This is because ratio data must have a meaningful zero and the lower levels of data may do not even approximately satisfy this requirement.

In summary, up-conversions are occasionally useful to experienced MDS users and probably should be avoided by beginners.

<u>Down-Conversions</u>

Down-converting attribute data is analogous to precision degradation of dissimilarity data. They are much more interesting, and more useful, than up-conversions. They produce rigorously correct results, even though the results do not exploit the full potential of the information contained in the data. Basically, down-conversions involve ignoring some information content of the data. The technique possibly provides a way of strengthening your conclusions. We will discuss this more in the following section after the nuts and bolts of Permap's conversions are described.

Downgrading from ratio level A_{ik} data to interval level data requires no action on Permap's part. You just start using the interval attribute-to-dissimilarity formulas. All the interval level formulas work off differences in attribute values and thus the attribute's zero point is irrelevant.

Downgrading from ratio or interval level A_{ik} data to ordinal level is done by first sorting the data into an array with the smallest values first. Ties are broken arbitrarily. Then, the first data group is defined to contain the entry with smallest value, or the first several entries where each value is less than or equal to 0.0001 more than the lowest value found in the group. Next, the previously grouped values are ignored and the grouping process is repeated. This continues until all entries are assigned a group. Finally, each attribute value is exchanged for the group's identification index (rank).

In coercing ratio, interval, or ordinal level A_{ik} data to nominal data the original data are rounded to the nearest integer, and the integers are treated as if they are names. The new "nominal" data will exhibit a greater coarseness than the original data. The degree of degradation is dependent on the range of the original data (you can increase the range by multiplying the data by a constant). If the original data are all integers, for example, there is no information loss. However, the map still will change because different types of attribute-to-dissimilarity formulas are now used.

Finally, ratio, interval, ordinal, or nominal level A_{ik} (remember that the nominal names are represented internally by numbers) data can be coerced into binary nominal data. There are two common ways to make the change and the choice is made using the Conversion to Binary option box at the bottom of the left-hand panel of the Dissimilarities-from-Attributes selection screen. The first option is to treat any zero value as binary zero, and any non-zero value as a binary one. The second is to treat any value less than or equal to the average value as binary zero, and all others as binary one. The average value is the average over all possible objects of the attribute in question. That is, it does not change if some objects are removed from the active data set by parking them.

6.1.2 Examples of Down-Conversions

Here are several examples of down-conversion to illustrate the rules given above. If your original attribute values are 0, .3, .6, 1.2, 55, and 11, they will become 0, 1, 2, 3, 5, and 4 when converted to ordinal level, and become 0, 0, 1, 1, 55, and 11 when converted to nominal level. If converted to binary nominal level, they will become 0, 1, 1, 1, 1, and 1 if the first metric-to-binary option is used, and 0, 0, 0, 0, 1 and 0 if the second option is used.

If your original attribute values are 0, .3, .6, .2, .1, and .5, they will become 0, 3, 5, 2, 1, and 4 when converted to ordinal level, and 0, 0, 1, 0, 0, and 1 when converted to nominal level. If converted to binary nominal level, they will become 0, 1, 1, 1, 1, and 1 if the first metric-to-binary option is used, and 0, 1, 1, 0, 0 and 1 if the second option is used. Clearly, the range of the metric data can play an important role in the conversion process.

6.1.3 Rational for Using Down-Conversions

Down-conversion is a valid procedure that should be understood by all MDS analysts. MDS is a very robust technique and sometimes down-conversion does not significantly change the final map. In these cases, your conclusions are strengthened because they are seen to be subject to less restrictive assumptions about data quality than they originally were.

For instance, say your attribute data consists of the averages of the responses of a group of tribal elders' judgments about the importance or occurrence frequency of a list of factors associated with numerous tribal ceremonies (objects). You have recorded all sorts of factors (attributes) and had the tribal elders use a Likert-like scheme where the bottom category of each factor's range was of the form "zero importance," "little importance," and so forth. Furthermore, you have arranged the check boxes in a manner so that there is an equal change in factor value between boxes. Based on your abundant confidence in your survey construction expertise, you tell Permap to assume that your data are ratio level and make the MDS map using COS(alpha) attribute-to-dissimilarity function (most likely a mistake but that is beside the point here). However, your research advisor says you cannot do that because you don't know that a checkmark in the bottom box truly means that the value of the factor in question was zero. For instance, maybe the tribal elders classified very unusual events as being of zero importance just because they were rare events. You disagree with your advisor. Instead of arguing the point (smart move) you reinterpret your attribute data as interval level data and redo the analysis (i.e., just click the Attributes shortcut button) using a less restrictive formula. Your results likely will not change at all and as a result you are in a stronger position to support your conclusions. Now, say your advisor further complains that you do not truly know that the intervals between the Likert levels represented by adjacent check boxes are equal. Again, you should reinterpret your attribute data as being ordinal level and redo the analysis (i.e., just click the Attributes shortcut button). Your MDS map may or may not remain stationary.

This line of reasoning can be extended to include the radical case of degrading your supposedly ratio level attribute data all the way down to the binary nominal level. For instance, a doctoral thesis examiner once was observed exclaiming that a candidate's data were so unreliable that they should not be called "measurements" at all. He stated that they were only indicators of the existence or nonexistence of the event under study. By now you should know what the candidate should have done before his thesis defense. He should have reinterpreted his data as binary nominal level data and checked to see if it mattered to the resulting MDS map. Depending on the nature of the data, and critically on the number of data points, the resulting map may not change at all and the candidate could have avoided a most undesirable confrontation.

Can any real data set undergo such severe down-conversion (precision degradation) and still yield the same MDS map? Many cannot but some can. Sometimes a very great amount of attribute data degradation causes almost no change in the MDS map. Whether or not the map is immune to precision degradation depends on the dimensionality of the data's hidden structure. If there really is a low-dimensional structure in the data, then the dissimilarity data are highly redundant and the results will be insensitive to some measurements being eliminated or having their precision degraded. This redundancy is expensive to the analyst in terms of the effort needed to produce the dissimilarity matrix, but it is beneficial when it comes to the analysis stage.

It should be clear that no magic is involved in the robustness of MDS results. Robustness should be expected whenever the MDS map shows a low stress configuration. The dissimilarity table has N (N - 1) / 2 values describing object-to-object relationships. However, if the data are well fit in two dimensions, then no more than 3N-6 (for N>2) object-to-object relationships are needed to construct an equivalent map (translations, rotations, and reflections are ignored). So, for example, if there are 15 objects then of the 105 dissimilarity values only 39 are really needed. Furthermore, if you know, say, that the objects are controlled by the value of two attributes, then even less information is required. In this case no more than 2N-2 data points are needed to fully specify the map and only 28 of the 105 original points are needed. Because of this great amount of redundancy it should be no surprise that data containing two-dimensional hidden structures can suffer significant degradation, or contain a significant amount of random error, or have many missing values, without the map significantly changing. That is one reason why MDS is so liked by so many data analysts.

We have discussed only one of two ways that the effect of uncertainty in attribute data should be evaluated. In addition to considering changes in attribute information level (i.e., ratio, interval, …) one should consider each attribute’s value imprecision and determine how this imprecision might affect the MDS map. This second approach might involve assigning error bounds to each attribute value and then using the rules of the theory of error analysis to flow these errors through to the resulting dissimilarities and then to the object positions on the map. This is almost never done. Actually, it is probable that it has never been done. To get useful results one would have to assign probability distributions (using ranges yields unreasonably pessimistic results) for the uncertainties present in each attribute value. Obviously, this approach is very difficult for problems of any significant size and we know of no MDS program that supports this kind of sensitivity analysis.

6.2 Attribute Functions

The Attribute-to-Dissimilarity Functions box lets you pick a particular attribute-to-dissimilarity function that is consistent with your data type. The only advantage to making a selection using this screen rather than using the main screen's Attributes shortcut button to cycle to your preferred choice is that the text box on the right side gives the mathematical formulas that are used. This assumes that you recognize i and j as object subscripts, k as an attribute subscript, that |x| represents the absolute value of x, that Rngk = Range of A_k = MAX(A_k) - MIN(A_k), that r = the product moment correlation coefficient, and that N is the number of valid cases (which is NOBJECTS from your data file unless you have missing data or parked objects).

1. Ratio Level Attribute Data Required

Function	Formula
COS(alpha)	D_{ij} = 1 - Cosine(alpha)

Alpha is the angle between attribute vectors i and j. This is sometimes called the Angular Separation coefficient.

2. Interval Level or Better Attribute Data Required

Function	Formula
Euclidean (unnorm.)	D_{ij} = SQRT {SUM(all k) {$(A_{ik}$-$A_{jk})^2$ }/ N}
Euclidean (normalized)	D_{ij} = SQRT {SUM(all k) {$(A_{ik}$-$A_{jk})^2$ / RANGE$(A_k)^2$}/ N}
City Block (unnorm.)	D_{ij} = SUM(all k) { $\|A_{ik}$-$A_{jk}\|$ } / N
City Block (normalized)	D_{ij} = SUM(all k) { $\|A_{ik}$-$A_{jk}\|$ / RANGE(A_k)}/ N
Guttman	D_{ij} = (1 - mu2) / 2
Pearson Type 1	D_{ij} = (1 - r) / 2

The Pearson r is the usual product moment correlation coefficient. The Guttman mu2 monotonicity coefficient is less well known; it is sometimes used with ordinal analyses. Occasionally it is appropriate to use a Person Type 2 transformation using D_{ij} = 1 - |r|. This must be determined by the nature of your data.

3. Ordinal Level or Better Attribute Data Required

Function	Formula
Spearman	D_{ij} = (1 - r) / 2

The Spearman r is the same as the Pearson product moment correlation coefficient, except it is applied to the ordinal indices (ranks) instead of directly to the attribute data.

4. Nominal Level or Better Attribute Data Required

Function	Formula
Nominal SMC	D_{ij} = 1 - A / N

The count A is found by counting the times the attribute names (i.e., the assigned numbers given to Permap) of objects i and j are both are the same. N is the total number of valid cases. It equals the number of different attributes used to define an object unless there are missing attributes.

5. Binary Level or Better Attribute Data Required

Function	Formula
Jaccard	$D_{ij} = 1 - A / (A + B + C)$
Gower/Russell/Rao	$D_{ij} = 1 - A / N$
SMC/Sokal/Michene	$D_{ij} = 1 - (A + D) / N$
Hamman	$D_{ij} = 0.5 + 0.5 (A - B - C + D) / N$
Yule	$D_{ij} = 1 - (AD - BC) / (AD + BC$
Askin/Charles	$D_{ij} = 1 - MAX\{A / (A + C), A / (A + B)\}$

The counts A, B, C, and D are found by summing the occurrences of the four possible cases. That is, A is incremented each time the binary attributes of objects i and j are both are 1, B is incremented each time the first attribute is 1 and the second attribute is 0, C is incremented each time both are 0, and D is incremented each time the first is 0 and the second is 1. This is usually illustrated using a 2-by-2 contingency (association) table where A and B are the first row entries, and C and D are the second row entries. A + B + C + D = N, the total number of valid cases mentioned above.

6.3 Using Attribute and Dissimilarity Data

The last option in the Attribute-to-Dissimilarity Functions box is not often needed but is very useful in some special cases. This option is disabled (grayed out), and it does not appear in the Attributes shortcut button's rotation list unless you have both dissimilarity and attribute data in your data file. If you independently know dissimilarity and attribute values (most unusual, count your blessings), then it is important to know if they provide consistent results. By selecting the Use Provided Dissimilarities you tell Permap to use the dissimilarity data rather than calculate them using the attribute-to-dissimilarity formulas. Then, by clicking the Attributes shortcut button, you can see if switching back to using the attributes makes any difference.

6.4 Euclidean and City Block Attribute Normalization

The interval level Euclidean attribute-to-dissimilarity formula is used frequently. However, it and the City Block formula, come in several varieties each differing by their normalizing factor. Note that this is quite different from all the other formulas presented above. All of the other formulas are defined as either normalized quantities or their normalization is inherent (e.g., the "counting" formulas).

The key question in designing Permap was whether or not to offer the unnormalized (sometimes called "raw") forms of the Euclidean and City Block attribute-to-dissimilarity formulas. While the unnormalized forms can lead to mischief in the hands of a novice, they are important to the expert. For instance, if Permap were to insist on using only normalized forms, then experts would be prohibited from complete control over their analysis. They might want to, say, enter a set of data where each object has two attributes that are based on the same units of measurement but one has a range of 0.5 and the other 1.0. There may be good reasons why some experts want the differing ranges, and they do not want their data normalized by Permap. The experts are correct in insisting that they be able to control the treatment of their data. Thus, despite the possibly of misuse, Permap includes the unnormalized Euclidean and City Block forms.

It is also important to recognize that the literature contains several different techniques for normalizing the Euclidean and City Block formulas. The three most common methods are (1) dividing by the range, (2) dividing by the average absolute deviation, and (3) dividing by the standard deviation. All three approaches are valid and all three save those new to MDS from using input that results in D_{ij} values greater than one, which in turn forces the rescaling of all D_{ij} values back to a maximum of one and may result in a serious imbalance between the relative influences of the various attributes. In order to keep the Permap interface simple, and because the three normalizing techniques produce similar results, only the range-normalized equations are used.

Control over which form, normalized or unnormalized, Permap uses is achieved by using the option box in the middle of the left side panel on the Attributes-to-Dissimilarities selection screen. If the Unnormalized Form option is selected, then no changes are made to your attribute data and the unnormalized equation will be used. In this case it is solely up to you to provide properly scaled attribute data. On the other hand, if the Normalized Form option is selected, then the Euclidean and City Block formulas are normalized by range and you can enter attribute data based on any desired scale.

7.0 MAP EVALUATION

Harshman (1984) wrote an important paper addressing how to check the appropriateness of MDS results. He discussed the importance of not making an analysis until questions about the theoretical appropriateness of the MDS model have been considered. That is, is there reason to believe that the relationships between objects should be properly represented by distances? If not, then MDS may not be the appropriate tool for the analysis. Torgerson (1958, 1965) also addressed this point. These papers should be the starting point for those new to MDS analysis.

Once you have a map, you must address the question Kruskal (Schiffman, Reynolds, & Young, 1981) asked, "How well do we know what we know? How do we spot warning signals -- and bring them effectively to the attention of first-time users?" While the situation has improved since 1981, it is still true that MDS maps often are misinterpreted and artifacts are mistaken for relationships. It is important to qualify a map before undertaking the painstaking effort of defining and interpreting groups and assigning dimensions.

Several map evaluation tests are given in Sections 7.1, 7.2, and 7.3. All of these tests have to do with the validity of the two-dimensional representation of multi dimensional data. They do not address the interpretation of the map.

7.1 Objects Evaluation Screen

If some object has a high relative contribution to the value of the objective function or to its overall out-of-plane tendency (to be defined in the following paragraphs) then there is reason to believe that its placement on the map does not result from a two-dimensional association with its neighbors. This can be tested by examining a Pareto Plot of the objective function components, or by two other ways that involve superimposing information on the map. Each option has context-sensitive information provided in a "Notes" text box at the bottom of the Map Evaluation/Objects Evaluation screen. The Evaluation screen can be reached by using the menu system or by the ctrl-O shortcut.

Printing a map with the evaluation markings on it is facilitated by having these markings persist after the evaluation screen is closed. See the Section 6.4.1 for the various printing techniques. After closing the evaluation screen if the Start or Continue button is pressed, or an open space is clicked, all evaluation markings are removed and Permap operation is returned to its usual mode.

7.1.1 Pareto Plot

A Pareto plot is a bar chart showing some important quantity on the Y-axis and its associated name, index, type, etc. on the X-axis, where the events on the X-axis are presented in the order of decreasing Y values. To MDS users the contribution to the objective function, or the out-of-plane tendency, is the quantity of interest.

Pareto plots are used in several disciplines. They serve to keep focus on the most important events and give an indication of the nature of the distribution of some important characteristic. The Pareto Principle, sometimes called the 80/20 rule, is often stated as "80% of the results (profits, complaints, fines, etc.) come from 20% of the sources (products, customers, accidents, etc.)." Sometimes it is stated in the inverse format, but regardless of statement format it emphasizes that event frequency is not uniformly distributed across causal roots.

In passing, those new to MDS might benefit from a few comments on a related point. The Pareto Principle forms part of the foundation of MDS. Were it not true that many situations are controlled by the "significant few" rather than the "insignificant many" factors (another way of stating the Pareto Principle) it would make little sense to use MDS at all. This is because nobody can meaningfully interpret or communicate the results of a structure in four or more dimensions, and without interpretation and communication there is little value in making the analysis. So, one can say that MDS users have faith that their data has a significant hidden structure and that it can be substantially described using two (three, in rare cases) dimensions. No theory says that this must be so, but we proceed as if it is so. Fortunately, a vast amount of experience suggests it is so in many cases.

It is useful to identify outliers on the Pareto plot. To do this, Permap adds average, 1% probability, and 5% probability lines. The one-tail critical probability lines are set at the mean plus 2.33 standard deviations and at the mean plus 1.65 standard deviations. These lines show, approximately, the values where the likelihood that an occurrence is due to random chance is less than one percent, and five percent, respectively. It is important to note that the classification resulting from these lines is nothing more than a useful fiction because no evidence has been given that the underlying distribution is normal (i.e., a Gaussian or bell curve distribution). More information on this is given in the next section.

7.1.2 On-Map Diagnostics

In addition to a Pareto plot, the Objects Evaluation screen lets you evaluate the map's two-dimensional fit by superimposing information on the mapped objects. Two kinds of information can be displayed on the map. The first is in terms of contributions to the objective function and the second is in terms of out-of-plane tendencies.

7.1.2.1 Objective Function Contributions

Each object's contribution to the objective function can be displayed three different ways. First, the range of all objects' contributions to the objective function can be split into quarters, and each object colored according to its place in the range. Black objects are the worst. They are in the highest quarter-range in terms of driving up the objective function. The second highest quarter-range is colored light gray, then gray, and then clear. Sometimes a quarter range is empty. For instance, if all objects contribute only V_1 or V_2 amounts where $V_1 > V_2$, then the objects contributing V_1 are black and the rest clear.

Second, outliers can be identified. This approach is useful if you do not care if an object's contribution is above or below the average, but instead care whether or not an object's contribution is significantly different from those of all other objects. Making the determination of statistical outlier status is done using the usual one-tail rules that if a value is greater than 2.33 standard deviations above the mean then the chance that its occurrence is due to random variations is less than one percent, and if it is 1.65 standard deviations above the mean then the chance is less than five percent. The "one percenters" are colored red and the "five percenters" are colored black. It is important to note that this classification, while very common, is just an approximation because no evidence has been given that the underlying error distribution is normal. The use of Chebyshev inequalities would produce more rigorous results, but they tend to be overly conservative and are never used to our knowledge.

Third, the relative value (i.e., when all values are rescaled to a zero-to-one range) of an object's contribution to the objective function can be superimposed on the object symbol. If the map is uncluttered, this is a useful technique, but the first two techniques quickly gain superiority as the number of objects in the analysis increases.

7.1.2.2 Out-Of-Plane Tendencies

Instead of considering an object's contribution to the objective function, you might ask about its propensity to leave the plane of the map when making a two-dimensional analysis. Analogously, what is its propensity to leave the line in a one-dimension analysis or leave 3D space in a three-dimensional analysis?

Asking about out-of-plane tendencies is fundamentally different from asking about the object's contribution to the objective function. It is possible that objects might be highly stressed (i.e., the minimum value of the objective function is large) even though their fit is not improved by making the map in a greater number of dimensions. For instance, consider a tetrahedron sitting on a plane where P denotes the object at the peak. Assume the dissimilarities between object P and the other three objects are maintained at equal values, and all are slowly decreased. At some point the four-object map fits perfectly (zero objective function value) in two dimensions. The tetrahedron will have become a planar figure. If the same three dissimilarities are further decreased then the stress will begin to increase but the configuration of objects will not want to distort into the third dimension. In this case the value of the objective function is not sufficient to describe the situation. What is needed is a measure of an object's out-of-plane tendency (OOPT).

OOPT can be calculated as follows, where we use terminology for the two-dimensional case. The analogous formulas for higher or lower dimensions follow immediately. Instead of writing the objective function just in terms of the planar coordinates (X, Y), add a perpendicular coordinate, Z, to the equation. Assume the badness measure is Stress, the distance measure is Euclidean, and that a ratio MDS analysis is of interest. Then, for any given object, one can show that the first partial derivative of the objective function with respect to Z is always zero when evaluated at Z = 0. The second partial derivative is not always zero, and therefore it controls the rate of change of the objective function at Z = 0. That is, the second partial derivative provides a measure of the tendency of the object to leave the X-Y plane if the two-dimensional map is slowly relaxed into a three dimensional map. So, $OOPT_i$ can be measured by some negative constant times the second derivative described above. A useful normalization constant is the sum over all i and j of W_{ij}. By convention, $W_{ii} = 0$. The end result is the following formula:

$$OOPT_i = SUM(all\ j)\ (W_{ij}\ (D_{ij}-d_{ij})\ /\ d_{ij})\ /\ SUM(all\ j)\ W_{ij}$$

This formula is strictly applicable for only ratio MDS using Stress and the Euclidean distance measure. However, it is a useful approximation when d_{ij} is determined by any Minkowski distance measure, even including the City Block measure. Moreover, if the D_{ij} term is exchanged for the appropriate error or rank bound, and $OOPT_i$ is noted to be zero if d_{ij} lies within both bounds, then it is useful for all types of MDS analyses. So, rather than develop separate OOPT equations for each MDS variation, we simply use the above equation. In effect, we shift from using a derived $OOPT_i$ relationship to using the above equation as a definition of $OOPT_i$.

Following the approach used for each object's objective function contribution, $OOPT_i$ is displayed in three different ways. First, by coloring the objects according to the quarter-range of all OOPT values that $OOPT_i$ falls in. Second, by calculating whether or not $OOPT_i$ is an outlier, as defined previously, and highlighting the 1% and 5% outlier objects. Third, by displaying the relative $OOPT_i$ value (i.e., $OOPT_i$ divided by the maximum absolute value $OOPT_i$) superimposed on the object symbol. See Section 7.1.2.1 for the procedures that are used to display these categories.

Finally, new users might be interested in a review of what to do if some objects have outlier OOPT values. Of course, the first step is to check all input data for calculation or transcription errors. Next, check to be sure the experimental conditions were appropriate for the outlier points. If these checks fail to resolve the situation, then you are faced with a more fundamental problem. Other than just "living with it" you have two choices. They are to segment your data set or move to a higher dimension.

Removing offending data points is disagreeable to most researchers. The tacit assumption is that if you just understood the underlying theory better, you would understand how to form a new construct that would bring the outlier into conformance with the other data. Even worse, an ethical question arises whenever data points seem to be arbitrarily eliminated. Nevertheless, sometimes it is exactly the right thing to do. Earlier, when describing the parking lots, an example about dogs and cats, versus a mixture of dogs, was used. You cannot know at the start of a study what you are going to find. And, the more refined your study, the more likely it is that you are going to discover unanticipated nuances. We recommend that you use the objects parking lot liberally to find a subgroup of objects that is well represented in two dimensions. Then, repeat the process on the objects that were parked. After both data sets are fully described, search for a theory that recombines them into a unified whole.

If data segmentation is not successful, then you should consider going to a higher dimensional analysis. Our advice is to go no further than three dimensions except with the most unusual cases. While it is tempting to do a little "fishing" in higher dimensions, it is rare that one finds experienced MDS users there. Coxon quotes Shepard as saying "If a solution exists, it probably exists in two dimensions; if it does not, then it certainly exists in three" (1982, p. 87). There have been many researchers that have discovered higher dimensional results that were never published because of the difficulties of fully communicating their nature. Remember, object groupings must be meaningful, reasonable, and interpretable or they are of no value regardless of the statistical fit that has been achieved. As Schiffman, Reynolds, and Young, (1981) point out, "dimensions that cannot be interpreted probably do not exist."

7.2 Links Evaluation Screen

The word "link" refers to any kind of connection or relationship between two objects. In this case, a link has two aspects: the distance between the objects, and the dissimilarity given for the objects. So, a link analysis involves inspecting the distances between objects in relationship to their underlying dissimilarities to determine if the map faithfully conforms to the underlying data.

The Links screen provides a Shepard plot and three on-map analysis options that examine the state of individual links. Each option has on-line context-sensitive information provided in a "Notes" text box at the bottom of the Links screen. This option can be reached using the menu system or the shortcut key ctrl-L.

Printing a map with the evaluation markings on it is facilitated by having these markings persist after the evaluation screen is closed. See the Section 6.4.1 for the various printing techniques. After closing the evaluation screen if the Start or Continue button is pressed, or an open space is clicked, all evaluation markings are removed and Permap operation is returned to its usual mode.

7.2.1 Shepard Plot

The Shepard plot is a very good way to determine the fit of an MDS map. For metric MDS it involves nothing more than plotting D_{ij} on the X-axis and d_{ij} on the Y-axis (some authors reverse the orientation) and observing how the points scatter about a 45 degree line. For ordinal MDS, it involves finding a monotonically increasing line that represents the d_{ij} vs. D_{ij} data, and then observing how the points scatter about this monotone line.

There are three important aspects to consider when using the Shepard plot. First, the data points should cluster about a central 45 degree line if making a ratio or interval MDS analysis, and about the center monotone line if making an ordinal MDS analysis. This clustering can be evaluated by R^2, the percent of the variance that is explained by using the central line as opposed to that explained by using the average. R^2 is the coefficient of determination. Second, there should be no outliers. This is checked by observing if any points fall outside two standard deviation bounds that are plotted adjacent to the central line. Third, there should be no evident pattern in the distribution of points. If these three tests are passed, then you have a good fit.

A technical detail of an ordinal Shepard plot involves describing the procedure for defining the central monotone line. We take the middle value to be the average of the d_{ij} values on the lower and upper monotone hulls. The upper monotone hull's points are found by starting with the lowest D_{ij} value (the first X-value) and defining its corresponding d_{ij} value to be the Y-value of the first point on the upper monotone hull. Then the next higher D_{ij} is considered. If its d_{ij} value is above that of the just previous Y-value, then it is defined to be the next Y-value on the upper hull. If it is equal to or less than the just previous Y-value, then the next Y-value is set equal to the previous Y-value. This procedure is repeated on up to the largest D_{ij} value, thus defining the upper monotone hull. A reverse procedure is used to define the lower monotone hull. The central line is obtained by simply averaging the upper and lower values. This central line provides a clear, easily calculated, representation of the ordinal solution.

Beginning MDS users might consider this advice. Making a good analysis of a Shepard plot is far better than comparing the objective function value to some critical value. In fact, the "critical values" of Stress quoted in the early MDS literature are not reliable at all. These values are fundamentally wrong, not just wrong in value. The key distinction is that while they are rigorous for showing object placement improvement for a given problem, they are not rigorous for showing the absolute merit of object placement between problems. For many years this was a sensitive point. Applied users wanted guidelines to quote to support their results and they perpetuated the use of certain critical minimum values of Stress to confirm that a map was valid. But, the more theoretically inclined researchers knew that the critical-values idea was fatally flawed. For instance, it has been shown that acceptable values of Stress vary with the number of objects, the number of dimensions, and the nature of the error in the data. These conclusions are supported by Spence and Graef (1974), Wagenaar and Padmos (1971), Isaac and Poor (1974), DeLeeuw and Stoop (1984), Arabie (1973), Borg and Groenen (1997, p. 38). Arabie (1978, p. 111) stated the point most directly in saying "Another point that in the interim has ceased to be arguable concerns the "good", "fair", etc. labels applied to Stress, irrespective of the number of stimuli, dimensionality, Minkowski metric, and other considerations. Those labels . . . have since been discarded by most users as unrealistic . . . ." The bottom line is, finding the minimum value of an objective function is very important and relative changes of the objective function between alternative mappings of the same data are important. But, the exact value of an objective function does not mean much. You are better advised to focus on the Shepard plot if you want to support your claim that you have a good MDS solution.

Despite the forgoing it is likely that you will need to quote some stress values to get your work published. It really doesn’t matter that these values are not reliable when a journal reviewer demands that they be shown and interpreted, and this is very likely to happen. Our advice is to focus on the Shepard plot and then simply state the value of SStress and either Stress or Stress1. As to their interpretation, you probably should quote Kruskal’s 1964 “rules-of-thumb” for Stress1 values are: 0 – perfect, 0.025 – excellent, 0.05 – good, 0.1 – fair, >= 0.2 poor” and then very briefly mention that these are rough estimates that are known to not be applicable in all situations. It will do little good to mention that Kruskal later indicated that these Stress1 values were not reliable quality indicators, or to mention that there have been many studies that show better critical Stress1 values FOR CERTAIN SITUATIONS, or to mention that there has been no study that shows a set of critical Stress1 values that apply to all situations. Our experience is that it is counter-productive to argue these points with those not well versed in MDS.

7.2.2 On-Map Diagnostics

7.2.2.1 Waern Links

Waern (1973) suggested using a cutoff value for discarding dissimilarity information to simplify the job of mapping the most important structure hidden in a dissimilarity matrix. She was able to reduce several complex similarity matrices to fairly simple maps by using only manual calculations. Her critical values often fell in the 0.45 to 0.65 range. Her method didn't preserve the metric level information in the data, but it did a good job of showing some of the hidden structure using only the simplest of procedures. As a consequence of Waern's work, maps that show only some of the dissimilarity relationships sometimes are called Waern link maps. Later, Kruskal and Wish (1978) suggested highlighting links with a similarity values greater than 0.3 to reveal "horseshoe" structures embedded in MDS plots. Even later it became common to use a cutoff value of about 1/3 and apply it "both ways." That is, first highlight the links in the lower 1/3 of all dissimilarity values, and then highlight the links in the upper 1/3 of all dissimilarity values. Using the "larger and smaller thirds" rule leads to two very simple Waern plots that test the basic plausibility of a solution.

When using the "smaller third" Waern plot, the connecting lines should not strike out across the map. They should fill in the small spaces between adjacent and near-neighbor objects. When using the "greater third" Waern plot, the connecting lines should criss-cross the map, connecting only the objects that are farthest from each other. These two tests, while admittedly elementary, still have merit. If these Waern plots don't make sense, then further analysis is not needed. It simply means that a low-dimensional representation of the multidimensional data is not adequate.

Finally, some analysts like to use Waern cutoff values different from the standard upper and lower 1/3 levels. Thus, Permap has a slider just below the Waern option buttons that allows resetting the cutoff to any value. Using small cutoff values can lead to unexpected behavior when highly symmetric data sets are being analyzed. This happens because many dissimilarity values are equal. Rather than make an arbitrary elimination of some links inside a group of links with equal dissimilarities, Permap shows all or none of a group of links that have equal dissimilarities. Thus, links with equal dissimilarities will appear or disappear all at once.

7.2.2.2 Stretched Links

A link's stretch is defined to be (d_{ij} - D_{ij}) / D_{ij}, its fractional displacement from its ideal length. This quantity would be called "strain" in the engineering disciplines, but since "strain" has special meaning to Psychologists, the more generic terms "fractional displacement," "fractional stretch," or less appropriately but easier to say, "stretch," are used here. Actually, as was the case with the badness functions that have singularities, Permap adds 0.0001 to the denominator to make the definition of stretch be (d_{ij} - D_{ij}) / (0.0001 + D_{ij}). Thus, Permap uses a pseudo fractional displacement formula. The word "pseudo" is dropped for the sake of brevity and is justified because the 0.0001 factor is inconsequential in terms of the visual display of the information.

For ordinal MDS, the stretch definition is modified to be (d_{ij} - Ld_{ij}) / (0.0001 + Ld_{ij}) or (d_{ij} - Hd_{ij}) / (0.0001 + Hd_{ij}), depending on whether the d_{ij} value is below (compressed) or above (stretched) its preferred value. See Section 5.2.2 for a definition of Ld_{ij} and Hd_{ij}.

The fractional stretch display provides the option of showing either all stretched links, i.e., those with mapped distances that are longer than D_{ij}, or just the links that are stretched so much that they are statistical outliers. If you choose to show all stretched links, only those with more than a 0.5% fractional stretch are actually shown. This is to keep from reporting links as being stretched when in fact their "stretch" is only "numerical noise."

Outlier links are identified using the same approach outlined for objects described in Section 7.1.2.1, except that you seek to determine if a particular link's stretch is significantly greater than what can be explained by random variation. This is done using the usual one-tail rules that if a value is greater than 2.33 standard deviations above the mean then the chance that its occurrence is due to random variation is less than one percent, and if it is 1.65 standard deviations above the mean then the chance is less than five percent. The "one percenters" are colored red and the "five percenters" are colored black. As with the object outliers, it is important to note that this classification is not rigorous because no evidence has been given that the underlying distribution is normal. It is, however, a useful approach for showing outliers.

While outlier identification is useful, it is important to remember its true meaning. It simply means that the identified link is significantly different from all the others. That does not mean the link is necessarily erroneous. If almost all dissimilarities are well fit, then a link with any stretch, even stretch due to trivial numerical noise, may appear to be an outlier. Conversely, if a set of objects is very badly fit then the standard deviation of the stretch factors may be very large and no particular link, even one that is greatly stretched, would be identified as an outlier. In both cases, the identifications are correct but misleading. Finally, don't forget that the outliers are determined by fractional deviations. The Shepard plot (see Section 7.2.1) shows deviations, not fractional deviations. Thus, it is possible to find that a link is identified as an outlier even though its placement on the Shepard plot is within the 1.65 and 2.33 standard deviation boundary lines, and vice versa.

Another word of caution is needed if you are using something other than Euclidean mapping distances. For instance, if you have chosen the City Block metric then all plots and stress calculations are made in terms of the City Block distance rather than the "apparent" distance showing on the map. This can lead to results that are correct, but "don't look right" when the map and the Shepard plot or map diagnostics are compared. Using the City Block metric can seldom be justified and beginners are advised to stick to the Euclidean distance measure.

7.2.2.3 Compressed Links

This option allows showing either all compressed links, i.e., those with mapped distances that are shorter than D_{ij}, or just the links that are compressed so much that they are statistical outliers. The details are identical to those described in Section 7.2.2.2 on Stretched Links, except the quantity (d_{ij} - D_{ij}) / (0.0001 + D_{ij}) is replaced by (D_{ij} - d_{ij}) / (0.0001 + D_{ij}).

7.3 Attributes Evaluation Screen

The Attributes menu provides an Attribute Fit plot and five ways to superimpose attribute gradient vectors on the MDS map. Each option has on-line context-sensitive information provided in a "Notes" text box at the bottom of the Attributes menu screen. If no attributes are given in the problem definition, then the Attributes menu entry is disabled (grayed out). The shortcut key for this option is ctrl-A.

Immediately below the Analysis Options box is a box that lists all available attributes. By highlighting a member of this box you determine which attribute is to be used for the Attribute Fit plot and for plotting the various on-map gradient vectors. If you have more than seven attributes you will need to use the right arrow to highlight the “missing” attribute numbers in the Attribute Selection List box.

Printing a map with the evaluation markings on it is facilitated by having these markings persist after the evaluation screen is closed. See the Section 6.4.1 for the various printing techniques. After closing the evaluation screen if the Start or Continue button is pressed, or an open space is clicked, all evaluation markings are removed and Permap operation is returned to its usual mode.

7.3.1 Attribute Fit Plot

The Attribute Fit plot is a scatter plot that shows how well one or all of the attributes are described by an optimally oriented uni-variant gradient running across the map. As usual, the coefficient of determination, R^2, is the key measure. A perfect fit results in $R^2 = 1$, whereas random behavior causes R^2 to approach zero.

Each object's attribute value is represented by a point with the actual attribute value, after being scaled to fit the zero/one domain, plotted on the X-axis and its projected value, using the best-fit gradient, plotted on the Y-axis. This scatter plot gives a clear visual method of determining the usefulness of a simple uni-variant gradient for a particular problem. While the X-axis values are scaled to fit the graph, if the best-fit is very bad the projected values may be outside the bounds of the Y-axis. These cases are shown with red half-circles at the upper or lower limits of the Y-axis. See the following section for a discussion of the principles involved in defining the gradient vector.

7.3.2 On-Map Diagnostics

This section describes various biplots that superimpose a second set of information onto an existing MDS map.

7.3.2.1 Attribute Gradient Vector

Once a map is made, if attribute values are known it is reasonable to ask about the distribution of the attribute's values across the map. In some cases the underlying distribution can be understood simply by printing the attribute values on the objects or by coloring the objects according to their attribute values. Permap offers both of these methods. However, if there is considerable scatter in the attribute values it can be difficult to determine the underlying pattern. In this case, the most important piece of information concerns the direction of steepest ascent of the attribute's values. This gradient can be shown by making a biplot which superimposes a vector on the map that points in the direction that the selected attribute increases most rapidly.

Finding the best possible gradient vector is easily done by using standard least-squares ideas. Because only linear uni-variant gradient fields are of interest, all parallel vectors are equivalent. Therefore, without loss of generality, we place the vector through the map's center point. The vector is defined by the angle theta, measured counter clockwise from the positive X axis.

Assume there is a linear attribute gradient across the map. Let the ideal attribute values be represented parametrically by the formula a + b t, where t varies from t = 0 at the vector tail to t = 1 at the vector tip. Actual attribute values, A_i, are known for each object i. The t-value, t_i, which corresponds to each $A_{i,}$ is determined by the perpendicular line running from the vector and passing through object i. One wants the ideal values to fit the actual values as closely as possible, so least-squares minimization is used to calculate the parameters a and b. That is, a and b are chosen to minimize the sum of the squares of the differences between a + b t_i and A_i for all i.

Having calculated a and b, one can calculate R^2 for the vector specified by angle theta. Then, theta is varied and R^2 is calculated for each value of theta. The ideal theta is the angle where R^2 is a minimum.

While the vector described above gives the "best possible" direction of a linear attribute gradient, this best possible value may be no good at all. For instance, if attribute values are randomly assigned to the objects there will still be a best possible gradient vector, but it is meaningless. The R^2 value is given in the text box at the lower right of the screen. It can be used to help tell whether or not the linear attribute gradient is meaningful.

7.3.2.2 Vector and Quartiles

In addition to showing the vector described in Section 7.3.2.1, this option uses the selected attribute's value to color each object. If a value falls in the upper fourth of the range of attribute values for all objects, then the object is colored black. Objects in the next lower quarter range are gray, then light gray, and then clear. This option is useful when the screen is congested and reading individual values is difficult.

7.3.2.3 Vector and Values

In addition to the vector described in Section 7.3.2.1, this option prints the value of the selected attribute on each object. This option is useful when the map has few objects that overlap each other. Use the View screen (Section 4.3) to set the object diameter to accommodate printing the values.

7.3.2.4 All Active Vectors

This option shows vectors for each of the attributes that were used to determine the dissimilarities that were used to make the map. This set of attributes is called the “active attributes” set.

Showing all vectors at once can help determine which attributes are redundant and which are fundamentally different from each other. For instance, if two or more vectors line up together (co-linear either in the same or opposite directions), then they are providing similar information and all but one are redundant for the given set of conditions. It might be that a composite variable (a new "construct" or a linear combination) would provide greater insight and show less random variation because of the averaging affect. On the other hand, if there are vectors that are ninety degrees to each other (orthogonal), then you have a good lead on assigning meaningful coordinates to the map. The best of all possible worlds is when, after your very first MDS analysis, all of your attribute vectors fall into two perpendicular groups!

7.3.2.5 All Available Vectors

This option is much like Section 7.3.2.4, except it includes all attributes, even those that were not used in making the map.

This option facilitates doing "grounded research" where, for instance, you want the best possible synthesis of all available data, and you are willing to give up generalizability in order to squeeze the last bit of information out of your data. If you have a low stress map with a good interpretation, and you are interested in evaluating additional attributes, then use this option.

7.3.2.6 Vector Tie Lines

To understand the construction of the gradient vector, it can be helpful to see the connections between the object's attribute values and the gradient as represented by the best-fit regression values assigned to the vector. This information can be revealed by checking the "Show Vector Tie-Lines" box. A perfect fit would have all tie-lines running perpendicular to the gradient vector.

It is possible that the tie-lines will fly off the map. Some tie-lines may run almost parallel to the gradient vector. This behavior is not due to an error in Permap. It can happen when the best possible fit is a very bad fit (say, $R^2 < 0.1$). While the domain of the independent regression variable is constrained to the map, if the attribute data are poorly fit there is no reason that the dependent regression values must be within the boundaries of the map.

7.4 General Comments Concerning Map Evaluation

Almost all experts in the field agree that MDS results should not pass or fail depending on some particular value of some particular statistic. The first consideration is, do your results have face value? This means that they should "make sense" to an intelligent but non-expert observer. The lack of slavish obedience to statistical tests works both ways. Weak statistics should not necessarily condemn meaningful results, and strong statistics should not be used to prop up frivolous results. All results should pass the "adequacy, interpretability, stability, meaningfulness, reasonability, and generalizability" hurdles, and if they do then they should be accepted.

These ideas are not new. Torgerson (1965), and Law, Snyder, Hattie, and McDonald (1984) strongly emphasized that adequacy and interpretability should dominate the debate over the appropriateness of a MDS result. They clearly stated that it is the researcher's responsibility to understand and rationally probe into the nature of the data, and that the researcher must not be enslaved to lesser indicators such as Stress and R^2. In the end, if a conflict exists between measures of statistical fit and a meaningful interpretation, the measures of statistical fit must yield.

8.0 CONVERGENCE

Selecting the Convergence menu allows you to control two parameters that are seldom changed. They control operations that are going on behind the scenes.

8.1 Convergence Rate

It can happen that you get a problem whose solution is unstable from a numerical standpoint. This seldom happens with "real" data sets containing a reasonable number of objects, but it is fairly common for test cases with highly symmetrical or uniform dissimilarities or for very large data sets (say, larger than 200 objects). In such cases you can increase the stability of the solution process by changing the iteration step size. This increased stability comes with a decrease in solution speed. Moreover, it does not always work. About all one can say is that it is worth a try if you happen onto a "bad actor" matrix or if you really have to treat hundreds of objects at a time (a questionable requirement). See Section 11.0, Solution Difficulties, for additional ways to increase stability and for the difficulties associated with making MDS analyses of very large data sets.

8.2 Convergence Precision

Convergence precision is normally not a concern because the solution algorithm continues to run while you are studying the map and the map's precision quickly becomes more than even the most demanding analyst wants. However, if you are using the Auto Start and Auto Repeat options to find multiple solutions starting from random initial positions, you may want to control the convergence precision in order to control the speed with which you scan many solutions. This can be done by setting the convergence limits using the Convergence menu. Permap offers three convergence limits. The first allows fast screening of data, the second provides a balance between speed and precision and is adequate for most situations, and the third is so demanding that it will usually exceed all reasonable requirements.

It is possible for solutions to pulsate. That is, the field of objects will slowly expand and contract and the objection will slowly rise and fall. This happens because Permap optimizes in a step-by-step manner and it takes a few iterations for all parts of the problem to be equally optimized. When this happens, if the convergence precision is set too low, it is possible to record fictitious values as being the best-found values. Be sure to use a high level of precision when finalizing your work.

9.0 SPECIAL CASES

Data for some interesting special cases are built into Permap. They include geometrically simple cases that illustrate certain points. They are valuable for learning about MDS and for checking Permap's results against known solutions. They also show unintuitive results and non-convergent behavior. The F3 function key provides a shortcut to the Special Cases selection screen.

The prime entry point to this screen was moved to a top level menu item at the request of users, but this screen's content logically (according to Microsoft's menu philosophy) fits as a submenu under the top level File menu. Thus, an entry point is placed there too. The two paths lead to the same screen. So, the Special Cases screen can be opened using the File menu, the Special Cases menu, alt-C, or F3. That ought to keep almost everyone happy.

Some special cases involved dissimilarity or attribute data that are generated using a pseudo random number generator. Usually you will want a new set of random numbers generated each time you visit the Special Cases selection screen. However, upon occasion you may want Permap to use the same set of random numbers each time a special case is selected. The choice is made by using the Reuse Random Numbers check box at the bottom-left of the Special Cases election screen.

10.0 SOLUTION TECHNIQUE

How a program finds a configuration that minimizes the objective function is probably not of particular interest to you. However, you must have confidence that the program does what it says it does. That is, a "black box" (solution algorithm) is acceptable only if you have confidence that the black box works right. This confidence usually comes either from knowing that the algorithm has been used successfully for a long time or from testing it against many test cases where the correct answer is known. However, there should be no doubt about the fact that an algorithm is never proven to be correct by using these approaches. Only a mathematical analysis of the algorithm can do that. Unfortunately, the mathematical analysis of algorithms is so difficult, and there are so few mathematicians that specialize in this kind of analysis, that the rigorous approach has become an endangered species. Fortunately, Permap passes the two empirical tests.

For those interested in what goes on inside Permap, it uses an iterative method of solving the MDS minimization problem. Basically, the method is a heavily damped form of Newton's method applied to one object at a time. This approach is widely used for problems that have a limited amount of known structure. It has proven effective for every "real" data set tested so far, but in theory this approach does have limitations. The mathematically inclined can easily construct D_{ij} matrices that are unstable. In these cases the objects may precess, oscillate, or even enter the chaotic realm.

If you have a problem that shows signs of instability, you can do several things to help Permap find the solution. First, you can reduce the step size (Convergence menu). Sometimes this is sufficient to tame unstable cases. Second, you can apply strong mapping weights (Analysis Parameters/Mapping Weights menu). De-emphasizing the distant objects will often reduce chaotic behavior. This is particularly true for data sets that have very simple geometric arrangements (a straight line or a large matrix of objects fitting perfectly in two dimensions). After a stable solution is found, use the Mapping Weights shortcut button to rotate to the desired mapping weights. Third, you can use one of the more stable objective functions (e.g., SStress) to find the solution, and then use the Badness shortcut button to rotate to using the badness function that you really want. Fourth, if you are making an ordinal MDS analysis, first make a ratio analysis and then use the MDS TYPE shortcut button to rotate to an ordinal analysis. If none of these techniques work, and you are working with a "real" data set, please email the details to Ron Heady at ron@heady.us. It will be the first case he has seen and he will enjoy working with you on the problem.

11.0 SOLUTION DIFFICULTIES

Local Minima

The topic of local minima was touched on in Section 3.5.4 where the automatic control features were discussed. The topic is important. Difficulties associated with local minima were recognized by the earliest researchers (Kruskal, 1964). They have been noted by all the major contributors to the applied MDS literature, and continue to merit comments in current reviews (Everitt & Dunn, 1995). Still, the topic does not get the careful attention it deserves and some practitioners still discount its importance. Shepard (1972) noted that except in special circumstances local minima have not been a serious problem. Several other authors have repeated this conclusion. Even some recent reviews (Stalans, 1995) suggest that though configurations due to local minima often occur, "they generally are not drastically different from the global minimum solution." However, our experience is that false results due to local minima are not only common, they are very common. Moreover, sometimes they are significantly different from the lowest stress configuration. Borg and Groenen (1987, p. 221) back up this conclusion. They point out that "MDS algorithms usually end up in local minimum" and then go on to note that this tendency is even greater when nonmetric MDS is used. After using Permap, or any other interactive MDS program, we believe that you will see that this conclusion is correct. You might even come to wonder how the early MDS researchers, who had very limited computational facilities, no real-time interaction, and no built-in graphical capabilities, could be so confident that their results were indeed representative of global minima.

Most early MDS programs (and even a few currently available programs) contained subroutines that would choose smart (sometimes called rational) starting points to combat the difficulties associated with local minima. Torgerson (1952, 1958), published a method that could be used to find a good set of starting points. Twenty years later Arabie (1973) and Kruskal and Wish (1978) were arguing for an alternative approach. Both of their papers suggested that it would be better to use random starting points. Kruskal and Wish limited their advice to when a solution has a "suspiciously large" stress value, although they did not suggest how suspiciously large values should be recognized. But, regardless of this advice most MDS programs continued to use smart starting points. In our opinion, these smart starting points are a liability rather than an asset. We believe they should be avoided. This is because even though smart starting points often lead to finding a global minima, when they fail you do not know that they have failed. You simply get the same wrong solution over and over. Obviously, the alternative of using random starting positions requires far more computation. So, historically, the decision has been between being fast and usually right, or being slow and almost certain that you are right. This choice was difficult for many years. Fortunately, computer speeds now have increased to the point that the original argument is moot. As a result, the need to be sure of the nature of your solution dominates and most MDS programs now provide for easily solving problems using many different random starting points.

Because of the importance of the topic of smart starting points, we digress to offer new MDS users a bit of historical perspective. Arabie's (1973) comments against using smart starting points were not well received by many MDS researchers. For instance, Clark (1976) and Spence and Young (1978) criticized Arabie for not providing any systematic comparisons between multiple-start and rational-start approaches. Arabie (1978) refuted the arguments against using random starting points and explained the difference between having theoretically correct starting points and having state-of-the-art starting points. Nevertheless, the argument continued. In 1992, Davison (p.100) concluded that the multiple-start approach was ". . . expensive, time consuming, and largely untested in a systematic way." Even as late as 1997, Borg & Groenen (p. 222) were still concerned about the computationally intensive aspect of the multiple-start approach. As the body of MDS literature matured, however, Arabie's concerns proved to be valid and the debate amongst experienced MDS users now has faded away. However, Cox and Cox, in a draft of a journal submission (see the web site http://www.ncl.ac.uk/mds/) show that, at least in terms of beginning MDS users, the problem is still with us. They refer to misleading comments in SAS and SPSS manuals, and the fact that these programs do not make it easy to make the needed randomly generated replications. They conclude that beginning MDS users are still in danger of falling into this old trap. We agree and strongly recommend that you use Permap's Auto Repeat and Auto Stop features and make many tests before you conclude that you have a global minima. For first-hand experience with the problem, load the data in EXAMPLE_E.txt. These are the data Cox and Cox used to illustrate the ease with which local minima can be mistaken for global minima.

Degenerate Solutions

A note on degeneracy is in order. Over the years quite a few MDS articles have referred to the problem of avoiding degenerate solutions. A degenerate solution is a rigorous solution from a mathematical viewpoint but it is not meaningful in terms of the reality that is being modeled. Degenerate solutions generally take the form of all objects being in a straight line or clustered on top of each other. When first writing Permap we planned a subroutine to combat this possibility. However, we found that it was not needed. Evidently, some of the early MDS computer programs used either very susceptible algorithms or exceedingly unrealistic starting points. While Permap can produce degenerate solutions, they are very rare unless you make an effort to find them.

For teaching purposes, it is useful to be able to demonstrate the existence of degenerate solutions. Because Permap loads objects in a straight line, and because degenerate solutions often take the form of a straight line, demonstrating degenerate solutions is easy. After loading a new problem, click Continue instead of clicking Start. This causes Permap to start searching for a solution using the existing straight line positions instead of random positions. Because a degenerate solution is usually nearby, the objects may simply slide over into their degenerate solution positions. If you click Jiggle the solution will immediately evolve into a meaningful solution. But that does not really show the fragile nature of some degenerate solutions. To show just how unstable these degenerate solutions can be, click Stop and drag an object just very very slightly out of the horizontal line of objects. Then click Continue, and unless you have made an extremely small adjustment, the objects will immediately fly off to a more meaningful solution. Interestingly, this technique works with almost any distance function, badness function, and dissimilarity matrix. Another degenerate solution can be found by imposing extreme boundary conditions. After solving a problem close the constraining boundary while the solution continues to run. Do this by clicking the Boundary shortcut. If it is not present on the main screen then use the View menu to add the it to the list of active controls. As the circle shrinks toward a zero radius the objects are pressed together until they are superimposed upon one another. Then, with the solution continuing to run, open up the boundary. The objects will stay in their superimposed "solution." A jiggle will quickly show that this degenerate solution is meta stable.

Very Large Data Sets

Finally, it is important to discuss problems with a large number of objects. Everyone wants to treat all of their objects at once. That is natural, but not always a good idea. Permap's upper limit on the number of points was recently increased from 200 to 1000 because users were requesting it and because newer computers have so much RAM and such high CPU speeds that static memory can be used (to increase speed) without the operating system resorting to the slow "memory swapping" approach to memory management. But there are two difficulties with analyzing problems having many objects. First, large problems can become unstable because the solution is controlled by insignificant figures. This can be helped by using a small step size option in the Convergence menu, but when the problem gets really large even this may not be enough. It all depends on the nature of the data. But even if the stability problem is manageable, the "solution" almost certainly is only one of thousands (millions?) of local minima. Densely packed objects can take on almost any number of arrangements all of which have essentially the same objective function value. These differences must not be taken seriously even though they seem to be significant to the untrained eye. If you use many points you must shift to looking only for the "big blobs” and pay no attention to the fine detail. This is not a defect in Permap. It comes from the "curse of dimensionality." MDS programs that don't reveal as much of the inner workings as Permap have the same defect, it is just invisible. A better approach than treating all points at once is to find high concentrations of objects (“blobs”) and analyze each separately, and then, perhaps, replace them with archetypical objects based on the common properties of the objects in the center of the concentration. Then, conclusions are drawn from the arrangement of the archetypical objects. The key point is that the detailed arrangements between close objects can, and usually is, overridden by the requirements of the relationships of the many many distant objects.

12.0 Permap HISTORY

Permap was developed by Professor Ronald B. Heady of the University of Louisiana at Lafayette, and Professor Jennifer L. Lucas of Agnes Scott College. An early version is described in Heady and Lucas (1996). The current Windows-based version of Permap is a descendant of a DOS-based version which was written in 1993-94 using a Windows-emulating program

(VBDOS). In 2000, Permap was rewritten for the Windows operating system to further increase its interactivity, add new features, and remove certain memory restrictions.

Professor Tony Coxon and his classes have provided valuable input that has speeded along Permap's evolution. Many thanks to them, and to all the others that have sent in suggestions for improvement or words of encouragement.

13.0 GETTING HELP

There are three ways to get help. The first is to press F1 or enter the Edit/Edit Help File menu to read the brief on-line help file that was described in Section 4.2.2. This file is more of a short reminder list than a full summary of how to use Permap. It contains some of the operational sections of this manual. The second is to refer to this manual. The third is to email Ron Heady at ron@heady.us. He is more than happy to help you understand Permap's idiosyncrasies, but he cannot do much about specific hardware problems.

14.0 LEGAL NOTICE

APPENDIX I
DATA FILE KEYWORDS

To be recognized, a keyword must be the first entry on a line. Keywords are usually capitalized to make them standout, but any mixture of upper and lowercase letters is acceptable.

Keyword	Purpose	Example Use	Comment
NOBJECTS	Set number of objects	NOBJECTS=...	1000 maximum
NATTRIBUTES	Set number of attributes	NATTRIBUTES=...	100 maximum
TITLE	Set a title line to be shown on output	TITLE=...	65 char. max
MESSAGE	Set a message to be shown on output	MESSAGE=...	65 char. max
SUBTITLE	Set a subtitle line to be shown on output	SUBTITLE=...	98 char. max
SUBMESSAGE	Set a submessage to be shown on output	SUBMESSAGE=...	98 char. max
DISSIMILARITYLIST	Announce that dissimilarity data follow		
SIMILARITYLIST	Announce that similarity data follow		
ATTRIBUTELIST	Announce that attribute data follow		
LOCATIONLIST	Announce that initial positions follow		
WEIGHTLIST	Announce that weight data follow		
NA	A missing dissimilarity value	..., NA, ...	or ..., na, ...

The following keywords are of a different nature from those shown above. Normally, they are not used. In fact, if they are ever used it is almost certainly in a classroom. These keywords are used to control start-up details of the program. They override default values and force Permap to start with specific settings. Thus they provide a way for a teacher to provide a data file that also controls the type of analysis, or, for a "save solution" file to provide all information needed to immediately restart the analysis from the conditions existing when the file was saved.

These keywords are entered with a following equal sign and a number chosen from those shown after each option. For instance, if the data file has STARTBadnessFunctionNum=1, then Permap will start with the SStress badness option being selected

STARTMDSAnalysisTypeNum=0
    0 Ratio, 1 Ratio + Bounds, 2 Interval, 3 Interval + Bounds, 4 Ordinal
STARTBadnessFunctionNum=0
    0 Stress, 1 Stress1, 2 SStress, 3 Multiscale, 4 Fractional
STARTDistanceFunctionNum=0
    0 Euclidean, 1 CityBlock, 2 Minkowski
STARTMappingWtFunctionNum=0
    0 None, 1 Weak, 2 Linear, 3 Strong, 4 Adjustable
STARTAttributeFunctionNum=0
    0 COS(alpha), 1 Euclidean, 2 City Block, 3 Guttman
    4 Pearson, 5 Spearman, 6 Nominal SMC
    7=Jaccard, 8=Gower/Russell/Rao, 9=SMC/Sokal/Michener
    10=Hamman, 11=Yule, 12=Askin/Charles, 13 Dissimilarities in file
STARTAttributeDataTypeNum=0
    0 Ratio, 1 Interval, 2 Nominal, 3 Binary Nominal, 4 Unknown

STARTObjectIDMethodNum=0

0 Number, 1 Name, 2 None

STARTDimensionsNum=2

1 One Dimensional, 2 Two Dimensional, . . . , 8 Eight Dimensional

STARTConstraintRadius=1.0

A number from 0 to 1, indicating the radius of the constraining boundary

APPENDIX II
EXAMPLE INPUT DATA

The following provides an example of a working data file. To be readable by Permap this file must be stored in an unformatted text format. Therefore, if you want to actually run this data, copy and paste it into a word processor and then save the file in ASCII or ANSI format. Be sure that you do not introduce any "strange" characters into the text file (sometimes WordPerfect will add an invisible termination character at the end of a file, and this can cause trouble).

All lines that do not start with a keyword or a number are comment lines. Comments are disregarded by Permap.

This example uses data from Kaufman and Rousseeuw's book "Finding Groups in Data" that gives the subjective dissimilarities between eleven sciences as seen by fourteen postgraduate economics students from several different countries.

****************** Start Copying Just Below Here *******************

```
MESSAGE=Differences Between the Sciences

NOBJECTS=11

DISSIMILARITYLIST
Astr,  0
Biol,  7.87,  0
Chem,  6.50,  2.93,  0
CSci,  5.00,  6.86,  6.50,  0
Econ,  8.00,  8.14,  8.21,  4.79,  0
Geog,  4.29,  7.00,  7.64,  7.71,  5.93,  0
Hist,  8.07,  8.14,  8.71,  8.57,  5.86,  3.86,  0
Math,  3.64,  7.14,  4.43,  1.43,  3.57,  7.07,  9.07,  0
Medi,  8.21,  2.50,  2.93,  6.36,  8.43,  7.86,  8.43,  6.29,  0
Phys,  2.71,  5.21,  4.57,  4.21,  8.36,  7.29,  8.64,  2.21,  5.07,  0
Psyc,  9.36,  5.57,  7.29,  7.21,  6.86,  8.29,  7.64,  8.71,  3.79,  8.64,  0
```

****************** Stop Copying Just Above Here *******************

The data can be separated with space(s), a comma, or both.
If there are missing data, they should be entered as "NA" or "na."

The leading names shown above are optional, but if used they must start with a letter, must not start with NA or na, and must not contain non-alphanumeric characters.

If you want to force Permap to start an analysis using certain object locations, use the LOCATIONLIST keyword followed by a set of locations. When Permap writes out a solution to the PermapBestSolnFound.txt file, it puts these locations in the file. This means that you can use the output file as an input file and have the analysis begin exactly where you left off. The number of lines of data must equal the NOBJECTS setting. Leading plus signs are optional on attribute or dissimilarity data. If desired, they can be used to improve the readability of the input file. If you enter a location list, Permap will automatically set the number of dimensions used in the analysis to equal the number of dimensions implied by the location list.

```
LOCATIONLIST
 -0.5262  -0.0030
 +0.2297 -0.3264
 -0.0307  -0.4252
 -0.2926 +0.1243
 -0.1282 +0.5070
 +0.1829 +0.4083
 +0.4286 +0.4962
 -0.3222  +0.0142
 +0.2130 -0.4146
 -0.3328  -0.2349
 +0.5785 -0.1459
```

REFERENCES

Arabie, P. (1973). Concerning Monte Carlo evaluations of nonmetric multidimensional scaling algorithms. *Psychometrika 38,* 607-608.

Arabie, P. (1978). Random versus rational strategies for initial configurations in nonmetric multidimensional scaling. *Psychometrika, 43,* 111-119.

Borg, I. & Groenen, P. (1997). *Modern Multidimensional scaling: Theory and applications.* New York: Springer.

Clark, A. K. (1976). Re-evaluation of Monte Carlo studies in nonmetric multidimensional scaling. *Psychometrika, 41,* 401-403.

Cox, T. F., & Cox, M. A. (1994). *Multidimensional Scaling (Monographs on Statistics and Applied Probability, No. 59).* New York: Chapman and Hall.

Coxon, A. P. M. (1982). *The users' guide to multidimensional scaling*. Exeter, NH: Heinemann.

Davison, M. L. (1992). *Multidimensional scaling.* New York: John Wiley & Sons.

DeLeeuw, J., & Stoop, I. (1984). Upper bounds for Kruskal's stress. *Psychometrika 49,* 391-402.

Ekman, G. (1954). Dimensions of colour vision. *Journal of Psychology, 38,* 467-474.

Everitt, R. S., & Dunn, G. (1995). *Applied multivariate data analysis*. London: Edward Arnold Publishers.

Fitzgerald, L. F., & Hubert, L. J. (1987). Multidimensional scaling: Some possibilities for counseling psychology. *Journal of Counseling Psychology, 34,* 469-480.

Harshman, R. A. (1984). "How can I know if it's real?, A catalog of diagnostics for use with three-mode factor analysis and multidimensional scaling." In H. G. Law, C. W. Snyder, J. A. Hattie, & R. P. McDonald (Eds.), *Research Methods for Multimode Data Analysis*. New York: Praeger Press.

Hartigan, J. A. (1975). *Clustering algorithms.* New York: John Wiley & Sons.

Heady, R. B., & Lucas, J. L. (1997). Permap: An interactive program for making perceptual maps. *Behavior Research Methods, Instruments, & Computers, 29*(3), 450-455.

Hubert, L., Arabie, P., & Hesson-Mcinnis, M. (1992). Multidimensional scaling in the city-block metric: A combinatorial approach. *Journal of Classification, 9,* 211-236.

Isaac, P. D., & Poor, D. S. (1974). On the determination of appropriate dimensionality in data with error. *Psychometrika 39,* 91-109.

Kaufman, L., & Rousseeuw, P. J. (1990). *Finding groups in data: An introduction to cluster analysis.* New York: John Wiley & Sons.

Kruskal, J. B. (1964). Nonmetric multidimensional scaling: A numerical method. *Psychometrika*, *29,* 115-129.

Kruskal, J. B., & Wish, M. (1978). *Multidimensional scaling: Sage university paper series on quantitative applications in the social sciences.* Beverly Hills, CA: Sage Publications.

Law, H. G., Snyder, C. W., Hattie, J. A., & McDonald, R. P. (1984). *Research methods for multimode data analysis.* New York: Praeger Press.

Lee, S-Y., & Bentler, P. M. (1980). *British Journal of Mathematical and Statistical Psychology, 33,* 142-150.

Schiffman, S. S., Reynolds, M. L., & Young, F. W. (1981). *Introduction to multidimensional scaling.* New York: Academic Press.

Shepard, R. N., Romney, A. K., & Nerlove, S. B. (1972). *Multidimensional scaling, Volume I - Theory*. New York: Seminar Press.

Spence, I., & Graef, J. (1974). The determination of the underlying dimensionality of an empirically obtained matrix of proximities. *Multivariate Behavioral Research, 9,* 331-342.

Spence, I., & Young, F. W. (1978). Monte Carlo studies in nonmetric scaling. *Psychometrika 43,* 115-117.

Stalans, L. J. (1995). Multidimensional scaling. In L. G. Grimm & P. R. Yarnold (Eds.), *Reading and understanding multivariate statistics.* Washington, DC: American Psychological Association, 137-168.

Taylor, J. R. (1997). *An Introduction to Error Analysis*. Sausalito, CA: University Science Books.

Torgerson, W. S. (1952). Theory and methods of scaling: I. theory and method. *Psychometrika, 17,* 401-419.

Torgerson, W. S. (1958). *Theory and methods of scaling.* New York: John Wiley & Sons.

Torgerson, W. S. (1965). Multidimensional scaling of similarity. *Psychometrika, 30*, 379-393.

Wagenaar, W. A., & Padmos, P. (1971). Quantitative interpretation of stress in Kruskal's multidimensional scaling technique. *British Journal of Mathematical and Statistical Psychology, 24*, 101-110.

Waern, Y. (1973). Structure in similarity matrices: A graphic approach. *Scandinavian Journal of Psychology, 13,* 5-16.

Young, F. W., & Hamer, R. M. (1987). *Multidimensional scaling: History, theory, and applications*. Hillsdale, NJ: Lawrence Erlbaum Associates.