Vignette: MDS-GUI

Andrew Timm

Abstract

The **MDS-GUI** is an *R* based graphical user interface for performing numerous Multidimensional Scaling (MDS) methods. The intention of its design is that it be user friendly and uncomplicated as well as comprehensive and effective. This document accompanies the MDS-GUI and should be referred to for demonstration of introductory use. Some basic theory of Multidimensional Scaling is first discussed and then the capabilities of the GUI are briefly demonstrated with two different data sets. The first set of data deals with Morse-Code signals and is used to show how the MDS-GUI differentiates between categories of the data. The second data set focuses on the nutrition content of breakfast cereals and demonstrates analysis with the use of underlying variable axes.

Draft Version
July 24, 2012

1. Introduction

This document is a companion to the *R* based program called the MDS-GUI. The purposes of this Vignette are to provide introductory information on Multidimensional Scaling and the use of the MDS-GUI. It is recommended that any user that is either new to the MDS-GUI or new to Multidimensional Scaling read this document. The Users Manual also accompanies the MDS-GUI and it should be referred to for more information on navigation of the program and descriptions of features and areas. This document is laid out in such a way that it is assumed that the user is familiar with the content of the Users Manual.

1.1 Multidimensional Scaling

Like all ordination methods, the purpose of all the types of MDS is to provide a visual representation of a large data matrix in a low dimensional space. From a simplified point of view, MDS is used to provide a mapped, usually two or three dimensional, approximation of the patten of proximities found in a given set of data. This set of proximities is either in the form of dissimilarities or similarities between objects in the data. More technically, what Multidimensional Scaling does is to find a set of vectors in p dimensional space (where p has been predefined) such that the matrix of Euclidean distances among them corresponds as closely as possible to some function of the input matrix according to a certain criterion, most commonly Stress. Each vector is then treated as the set of coordinates of the corresponding dimension, thus allowing a visualisation in p dimensional space such that each object in the data is represented by a point on the plot. The distances between these plotted points represents, as accurately as possible, the original similarities (or dissimilarities) of the data. This implies that similar pairs of objects are represented by points that have been positioned closer to one another and dissimilar objects are represented by points that have been positioned further apart from each other. It is for this reason that Mair and Leeuw (2008) describe MDS as a set of methods for discovering "hidden structures in multidimensional data". For comprehensive source of information on Multidimensional Scaling, the use is referred to Cox and Cox (2001), Borg and Groenen (2005).
The MDS-GUI provides eight different methods of Multidimensional Scaling to the user. The Metric MDS options include: Classical Scaling, Metric Least Squares Scaling and Metric SMACOF. The Non-Metric Options include: Sammon Mapping, Kruskal's Analysis and Non-Metric SMACOF. Each of these methods are performed with the an $n \times n$ dissimilarity matrix Δ as the input. Finally, two alternative MDS methods are provided, being the Gifi model and INDSCAL. These two methods require an $n \times m$ data matrix $\mathbf{Z}$ as input, i.e. the data before being converted to a dissimilarity matrix. The MDS-GUI however handles all necessary input management automatically, and the user may simply choose their desired method.

1.2 Existing Software

Multidimensional Scaling capabilities are available in many mainstream analytical software packages, such as STATISTICA (statsoft, 2012) and the SAS software package (SAS Institute Inc., 2011). These suites are however not open source and require payment for licenses by the user. The packages are also not solely intended for Multidimensional Scaling and have been found to have steep learning curves. The MDS-GUI will be the first publicly available MDS specific performing GUI for the *R* environment. It is however not the only MDS user interface that is freely available to the public. Two open source programs that have been developed are the iMDS package for Matlab (Groenen, 2003) and the X/GGVis software (Buja et al., 2004).
The iMDS software is a prototype interface written in Matlab. The package includes features such as: dragging points in the MDS plane; allowing various transformations (interval, ordinal, monotone, spline); Shepard plot with brushing to identify pairs of points in the MDS plot; dynamic view of iterative process and setting weights as a power of their dissimilarities. The current version of iMDS (v0.1) does not allow for importing of ones own data. A few popular datasets have been included and the software is limited to the use of these. The iMDS package should therefore be seen as a functional means of demonstrating Multidimensional Scaling. The package is available for free download at http://people.few.eur.nl/groenen/.
The XGVis and GGVis software packages are designed to perform Multidimensional Scaling in a visual and

interactive way. They incorporate the already existing XGobi (Swayne et al., 1998) and GGobi (Swayne et al., 2002) packages as graphical engines. The program is very detailed with numerous functions. Some of which, as mentioned by Buja et al. (2004) are: Experimenting with various parameters; subsetting objects; subsetting dissimilarities; weighting dissimilarities; manually moving points and groups of points; perturbing the configuration or restarting from random configurations. XGVis is available for free download from www.research.att.com/areas/stat/xgobi and GGVis is available for free download at www.ggobi.org.

1.3 R

R is the name of a computing language that has become affiliated with data analysis and graphical representation techniques. *R* is a "GNU project", where 'GNU' is a recursive acronym which stands for "GNU's Not Linux" and represents a group of projects similar to Linux based systems but not affiliated to them. It is a an open source addition to the similar *S* language developed by John Chambers (Chambers, 2008), also one of the chief developers of *R* (R-Development-Core-Team, 2012). The *tcltk* interface that has been developed for *R* as well as *R* specific functions were used exclusively throughout the practical side of the project.

The *R* language is a common programming format for statisticians and the RGui is well known by the vast majority of those needing to perform statistical procedures on data. As with all open source pieces of software, the product is available for free download and is open for contribution by any author. This means that while the default functionality of an *R* program might be limited, especially when it comes to specialised applications, any member of the *R* community that has developed new features may make them available to all other *R* users. These contributed pieces of code are compiled into what are called "packages" and are available on the *R* website.

The base code of the *R* language was written and is maintained using the low level coding language called *C*. *R*, like *Tcl*, is considered 'high level'. As *R* has a strong relationship with *C*, many computationally intensive *R* processes can be written in *C* or *C++* and then called upon within the function. This sort of procedure is however very much considered to only be accessible to advanced users.

2. The MDSGUI package

The MDS-GUI will be available in the *R* package called **MDSGUI**. This package will contain only one function, being `MDSGUI`. This function will have no required input parameters and is simply utilised using by typing 'MDSGUI()' into the *R*-Console. At the time of development, the optimum version of *R* to use was R version 2.13.0. A drawback of incorporating external packages into a new package is that the package being developed is subject to the limitations of these packages. An example of this occurred when attempting to use the highly popular RStudio software (RStudio, 2012) where unexpected errors attributed to the **tkrplot** package. As a result, the current version of MDS-GUI is only compatible with the 32 bit version of the RGui (R Development Core Team, 2011), and it is suggested that the 2.13.0 version is used. In time these bugs may be seen to by the respective developers in which case these restraints are likely to be lifted.

2.1 The MDS-GUI

The MDS-GUI is called using the `MDSGUI()` command. No parameters are required to be input when using the function. Data IS NOT uploaded to the GUI upon start up and does not need to be in your *R* environment. Instead, data is uploaded to the GUI from an external file from the GUI itself, and may be in the format of txt or csv. Figure 1 shows the MDS-GUI as it appears when first loaded.

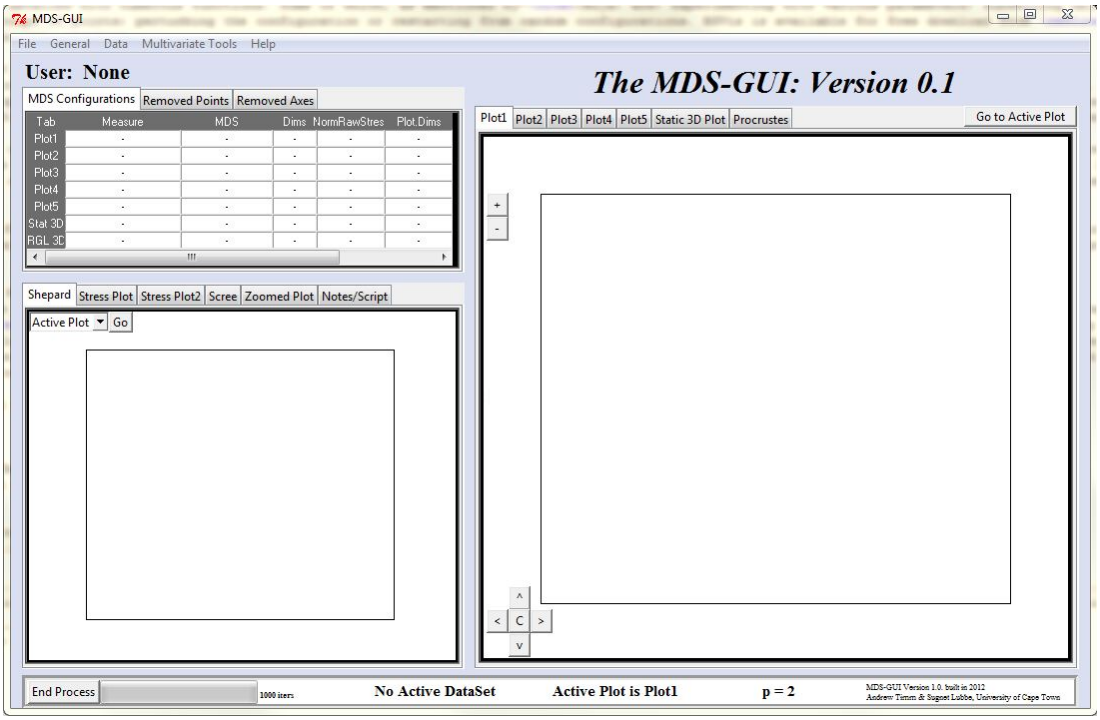

Figure 1: The MDS-GUI

3. An Example

3.1 The Morse-Code Data

The study done by Rothkopf (1957) involved the collection of confusion data from subjects identifying the audio similarity between 36 Morse code signals(26 letters, 10 numbers). The result of this was a 36×36 asymmetric matrix. This set of data has become a favorite for demonstrating Multidimensional Scaling procedures and can be found in many textbooks and papers on the subject. Examples include Borg and Groenen (2005), Buja et al. (2004), Carroll and Chang (1970), Maechler (2009), Everett (2001), among others. The inclusion of this particular data is due to its popularity, as results from the MDS-GUI may be compared to previous results for confirmation of accuracy. As with many MDS programs, the functions of the MDS-GUI require any dissimilarity/similarity matrix to be symmetric. The adapted symmetric version of the square similarity matrix (also provided by Rothkopf) is therefore used throughout this section. Each element of the matrix represents the percentage of respondents that determined the signal pairing to be the same.

The data that will be used by the MDS-GUI includes a column indicating the length of each symbol. For example ‘E’ has one element and ‘9’ has five. This categorical information will play an important part in this analysis.

3.2 Getting Started With the Morse-Code Data

Once the MDS-GUI is loaded, all actions and commands happen from the GUI itself. No coding is required in the console. To load data into the GUI, do the following.

1. The Morse-Code data initially comes in form of the **S** Similarity matrix, so this must be specified when loading the data. In the MDS-GUI, go to *Data* → *Load Dissimilarity Matrix*.
2. Navigate to the whichever folder your data file is held. In this case the selected file is called morsecodesymL.txt. The file is selected.
3. The *New Active Dataset Options* window will open. Figure 2 shows what this window should look like. Name your data however you wish. It is important to indicate that a categorical variable is present in the data.
4. The Morse-Code data has now been uploaded to the MDS-GUI and the dissimilarity matrix Δ has been calculated automatically.

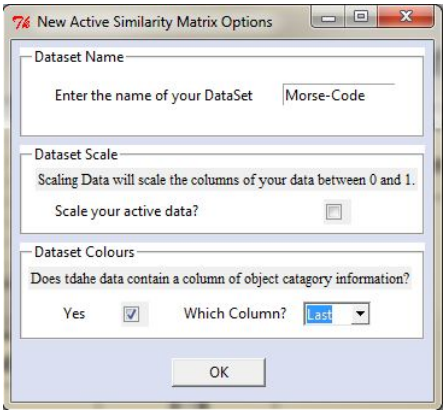

Figure 2: New Active Dataset Options: Similarity Matrix

3.3 Analysis of Morse-Code Data: A step by step beginners guide

The Morse-Code Data is now active in the MDS-GUI. The categories of this data is defined by the sequence length of each object.

1. Perform Classical Scaling: Do this by going to *Multivariate Tools* → *MDS* → *Classical Scaling*. By default p=2. The result is shown is in Figure 3
2. Observe:
   (a) **Configuration:** The MDS configuration, **X**:36×2, is shown in the main plot window of the GUI. Each of the 34 objects are shown with the distance been points indicating how similar they are. The colour of each point is defined by the group in which it belongs.
   (b) **Shepard Diagram:** The Shepard Diagram is housed in the *Shepard* tab of the plots in the bottom left of the GUI. Each point represents a pairing of points of the data, and thus 630 points are found on this plot. The X-axis corresponds to the observed distances, δ_{ij}, and the Y-axis to the MDS distances, d_{ij}. Points lying above the transformation line show the distance between the object

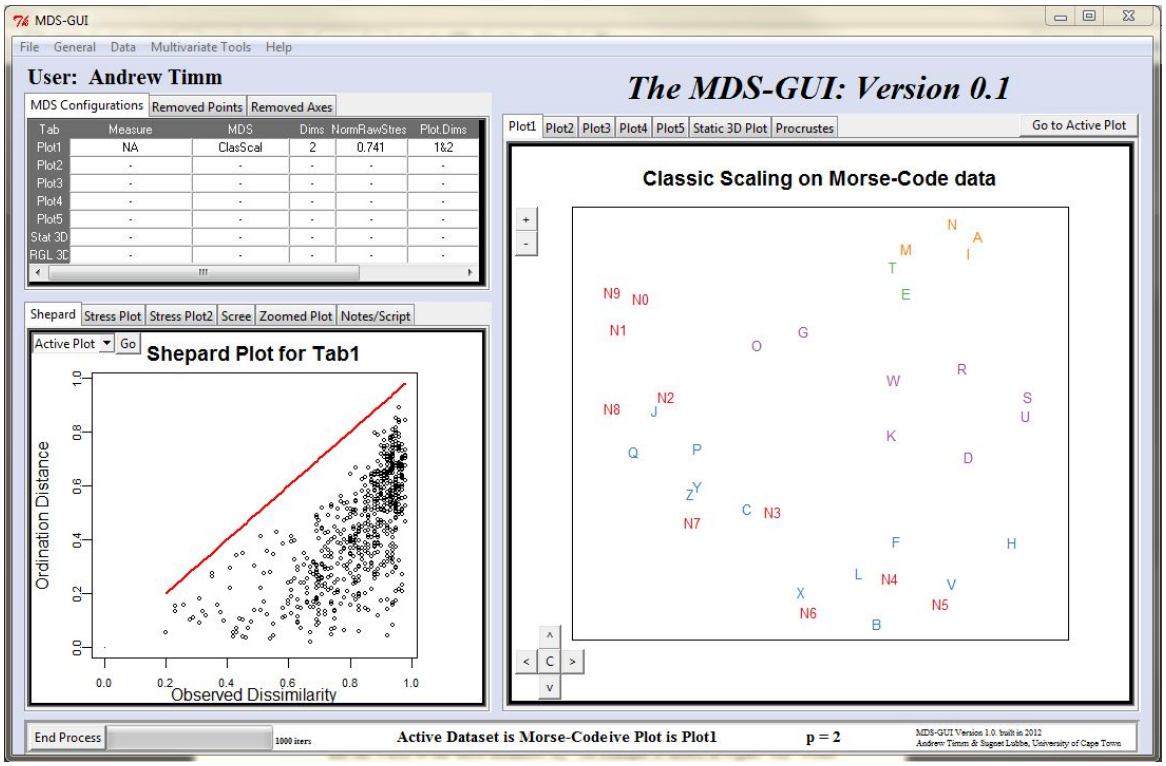

Figure 3: Morse-Code: Classical Scaling

pairing have been overstated by the MDS process and those below it have been understated. Clicking any point will identify the object pairing on both the Shepard Plot and the configuration plot. Brushing the plot will highlight all points selected.

(c) **Scree Plot:** The Scree Plot gives an idea of how many dimensions is most appropriate to the data in which to perform the MDS. It shows the change of stress over values of p. The plot shows the current dimension and optimum dimension according to the change of stress. The scree plot is housed in the *Scree* tab at the bottom left of the GUI.

(d) **Information Table**: The table shows all relevant information to the MDS procedure. Most importantly is 'stress' which indicates the goodness-of-fit of the process. The smaller the value, the better the fit.

3. Now try...

(a) **Adjust the category colours:** Go to *Data* → *Category Colours*. Select the colour square to change colour of the selected category.

(b) **Adjust plot appearance:** Right click the configuration plot to get the *Plot# Menu*. Then Choose *Plot Options*.

(c) **Utilise Plotting Areas 2-5:** The MDS-GUI allows for up to 5 simultaneous MDS procedures at a time. Selecting between the areas from *Plot1* to *Plot5* brings that area into focus. The Information Table allows for direct comparison between the results of each area.

(d) **Use Different MDS methods:** The MDS-GUI has eight MDS methods available for use. Try these from *Multivariate Tools* → *MDS*. Some methods, such as both SMACOF options, animate

the optimisation procedure. When this animation is happening, select the *Scree Plot* and *Stress Plot2* to observe the change in stress through the iterations. When using SMACOF, if the maximum number of iterations is reached, go to *General* → *General Settings* → *Convergence Tab*. From here you may either increase the maximum iterations of increase the tolerance value.

(e) **Drag Configuration Points:** By holding a left click over a point in the configuration plot and moving the mouse, you can drag the point around. Do this to observe how the Shepard Plot changes and how the stress value is effected as **X** changes.

(f) **Perform Procrustes Analysis**: When two of the plotting tabs have configurations in them, Procrustes Analysis may be used to illustrate the degree of similarity between the two results. To perform Procrustes Analysis go to *Multivariate Tools* → *Procrustes Analysis*. Then select the areas with the configurations to be compared. The result is shown in the *Procrustes* Tab.

After some exploration of the features of the MDS-GUI the result shown in Figure 4 can be reached. This shows the result of Non-Metric SMACOF on the Morse-Code data which produced the lowest stress value of all the available options. Analysis of the configuration clearly shows that each of the groups of different sequence lengths are defined and separated. A conclusion is thus that the subjects who took part in the study by Rothkopf (1957) were more inclined to incorrectly identify two sequences as the same when they were both of equal length.

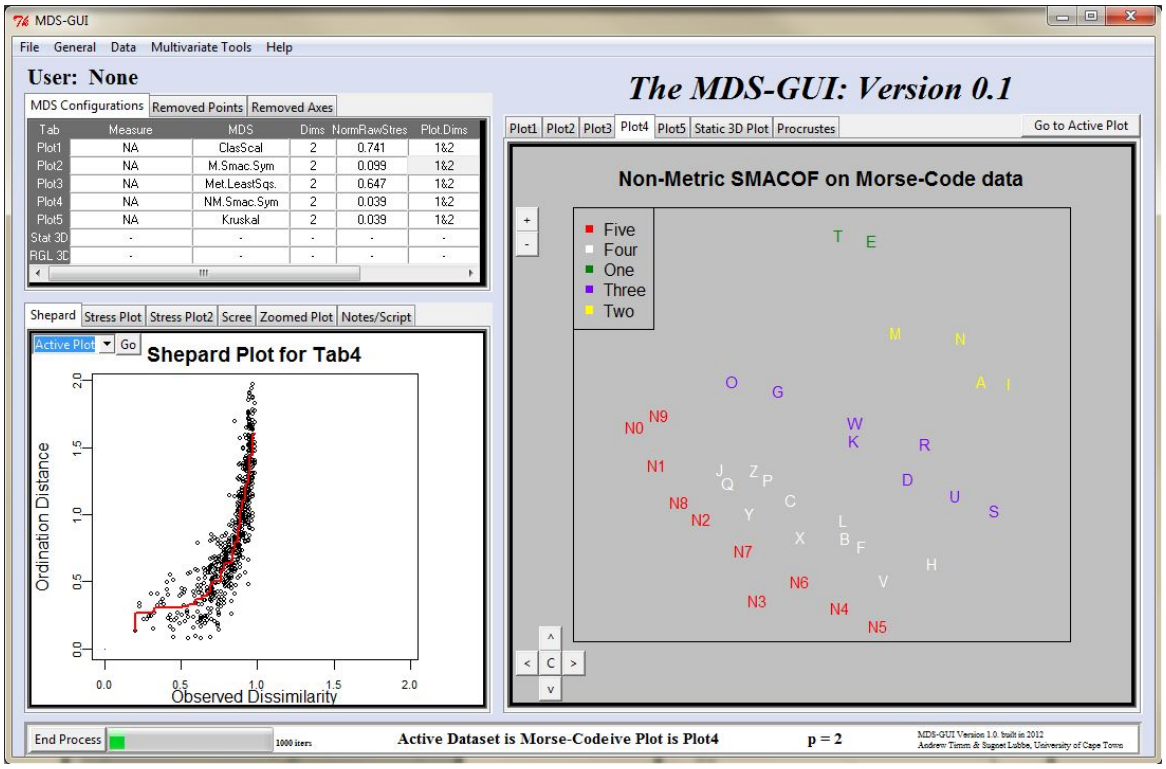

Figure 4: Morse-Code: Non-Metric SMACOF

4. Another Example

4.1 The Breakfast Cereal Data

The Breakfast Cereal data consists of 23 Kellogg's Cereals with ten different measurements made on each. Of the ten variable measurements, nine constitute various nutritional components and are all measured on the ratio scale, as significant zero's exist in each case. The last is a categorical variable and indicates the shelf of the store (1,2 or 3) on which the make was placed at the location of data collection. The relevance of the categorical variable is that it indicates the shop staff's perception of the association between makes. The main benefit of analysing a data-set such as this is in the analysis of the variable axes, with each of the ten variables having its own axis through the MDS configuration output. A similar Multidimensional Scaling analysis on this data was performed by Cox and Cox (2001). According to their suggestion, that scaling of the data is appropriate for MDS, the data will be scaled such that each column (variable) ranges between zero and one. This task is easily performed by the MDS-GUI. Upon uploading the data into the GUI, the researcher simply needs to select the Scale your active data check-box in the New Active Dataset options window. Alternatively, if an already uploaded set of data is in need of scaling, the same option is available in the Data Options menu.

4.2 Analysis of Morse-Code Data: A step by step beginners guide

1. **Upload the Cereal Data**: The breakfast cereal data is in the form of an $n \times m$ **Z** matrix. It must therefore be uploaded using the *Load Dataset* option. In the MDS-GUI, go to *Data* $\rightarrow$ *Load Dataset*. The new data window looks slightly different to the similarity version used before. The new window is shown in Figure 5. In this case, only a name and indication that the data is to be scaled is required. No categorical variable is identified here. The shelf variable will be treated the same as the others in this case.

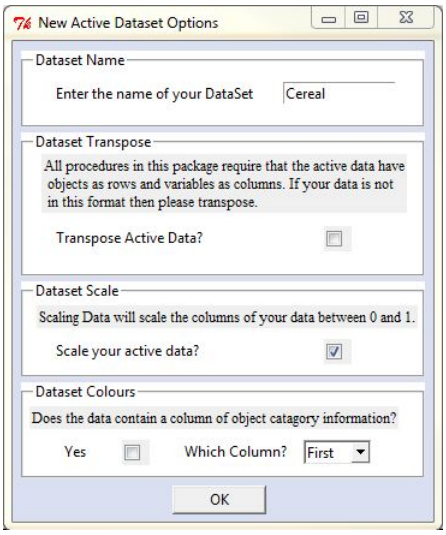

Figure 5: New Active Dataset Options

2. **Repeat Analysis as before**

3. **This time...**

   (a) **Experiment with Different Metric Calculations:** Now that an $n \times m$ **Z** matrix is used, the dissimilarity matrix Δ may be calculated in a number of ways. By default it is done using the

Euclidean Metric. To change the metric method used to calculate the dissimilarity matrix Δ, go to *Multivariate Tools* $\rightarrow$ *Dissimilarity Matrix Calculation* and select your desired method. Any MDS process performed from now will be using this metric (until it is next changed). Experiment with MDS methods on different metrics. Find which combinations produce lowest stress and most interpretable results.

(b) **Display Variable Axes:** This displays the m=10 variable axes through the origin (when $p = 2$). Interpretation of these axes is as follows. Each axes has a positive and a negative end, and object may be observed more or less influenced by a variable depending on how close they lie to an axes and towards which end they are affiliated. Additionally, the relationship between variables may also be observed. Axes running very close to one another are strongly correlated, with combinations in the same direction showing positive correlation and those running in opposite directions showing negative correlation. Axes that are perpendicular shown zero correlation to one another.

After exploration and experimentation, the result shown in Figure 6 can be achieved. The Kruskal's Analysis

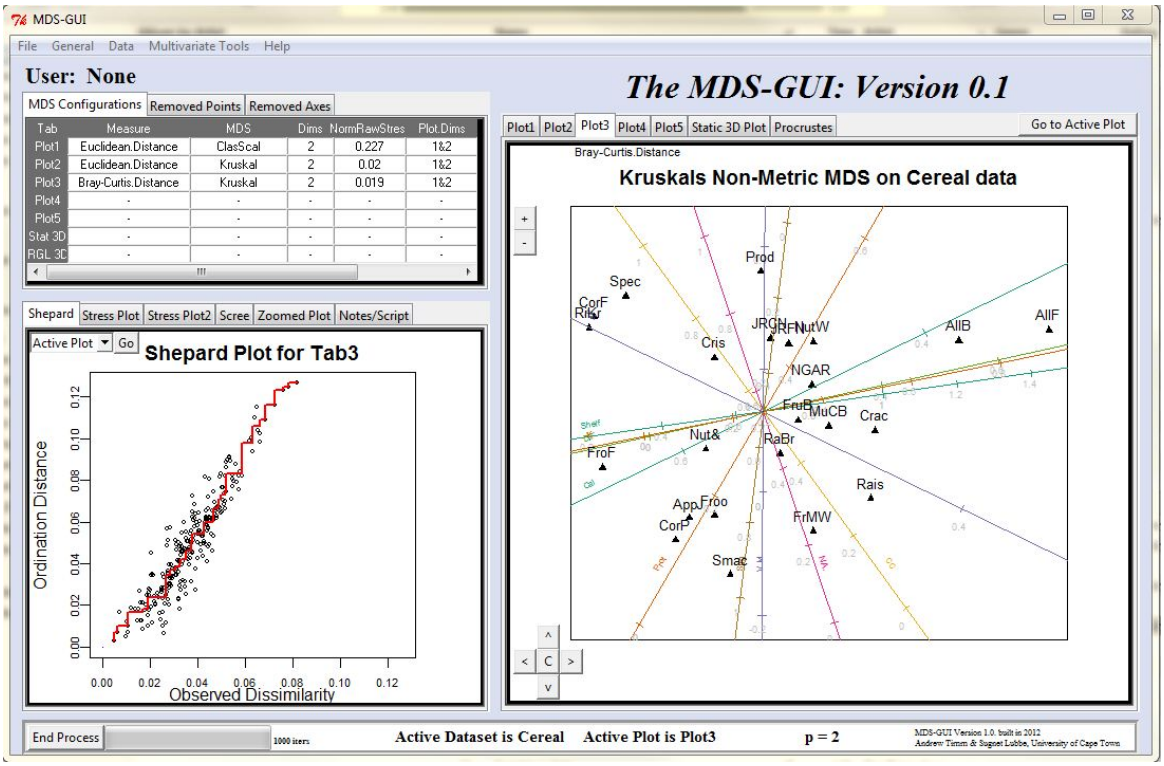

Figure 6: Cereal: Kruskal's Analysis with Bray Curtis Metric

method using the the Bray-Curtis metric as the dissimilarity measure produces a satisfying result. Of the many observations that can be made, the most interesting is the relationships between the *Dietary Fiber*, *Calories* and *Shelf Number* variables. These are placed such that the *Dietary Fiber* and *Shelf Number* axes run in the exact same direction indicating high positive correlation. Also the *Calories* variable is on the same line but runs in the opposite direction as the other two, indicating negative correlation. This result suggests that the cereals are placed on the shelves in the shop according to their perceived level of healthiness, as cereals with

higher fiber content are considered more healthy and those with higher calorie content are considered less healthy.

5. Also Try...

The MDS-GUI has many other useful features for MDS based analysis. Interested users may want to try some of them. For example...

1. **Three Dimensional Plots:** To change the number of plotting dimensions, *r*, go to *Multivariate Tools* → *MDS Options* → *Dimensions Tab*. All results will from this point be performed in the new number of dimensions. Choose 3 for 3D plots.

These remaining features may all be accessed through the right click menu on the configuration plot.

2. **Zoom:** From menu or by using the '+' and '-' buttons on the plot (or keyboard).
3. **Rotate and Reflect:**
4. **Move selection of points:**.
5. **Use Altered Configuration as Starting Configuration:**- Use SMACOF for best results.

5. An Important Note

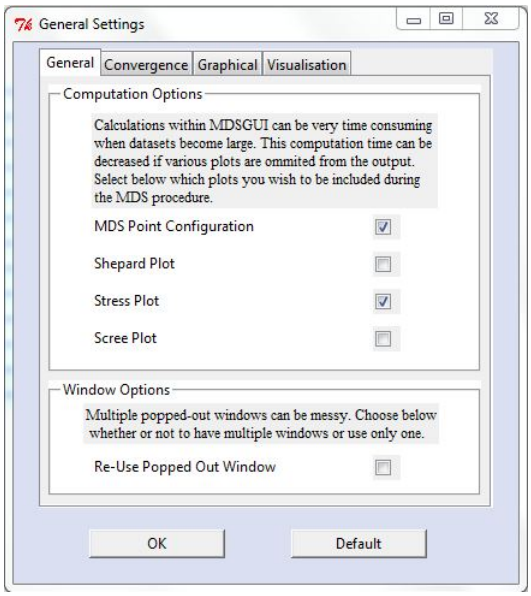

Figure 7: General Settings Menu

As with all analysis of data, the smaller the dataset, the faster the procedures may occur. The MDS-GUI often runs many simultaneous procedural calculations at the same time. As the size of the data increases, so does the time of all the processes. The most notable processes that experience greater lag as the size of the data increases is the calculation of the Scree Plot and all brushing processes. The biggest sized data that the MDS-GUI can handle on current computers without experiencing any lag is data with less than 60 objects. The MDS-GUI is capable of handling data with greater dimensions, however it is suggested that certain processes be deactivated. When a large set of data is loaded, go to *General* → *General Settings* → *General Tab*. De-select Shepard Plot and Scree Plot as shown in Figure 7. Click 'OK'.

Bibliography

Borg, I. and Groenen, P. F. (2005). *Modern Multidimensional Scaling: Theory and Applications Second Edition*, Springer, New York.

Buja, A., Swayne, D., Littman, M., Dean, N. and Hormann, H. (2004). *Interactive Data Visualization with Multidimensional Scaling*. University of Pennsylvania.

Carroll, J. D. and Chang, J. J. (1970). Analysis of individual differences in multidimensional scaling via and n-way generalization of "eckart-young" decomposition, *Psychometrika* **35**: 283–319.

Chambers, J. M. (2008). *Software for data analysis programming with R*, Springer, Berlin.

Cox, T. F. and Cox, M. A. (2001). *Multidimensional Scaling: Second Edition*, Chapman and Hal, Boca Raton.

Everett, J. E. (2001). *The Practical Handbook of GA, v1 Applications*, Chapman and Hall/CRC.

Groenen, P. (2003). Interactive multidimensional scaling iMDS v0.1. A standalone Windows application (XP, Vista) that allows dynamic form multidimensional scaling.

Maechler, M. (2009). *Interface to the XGobi and XGvis programs for graphical data analysis*. R package version 1.12.
URL: *http://CRAN.R-project.org/package=xgobi*

Mair, P. and Leeuw, J. D. (2008). Multidimensional scaling using majorization: Smacof in r, *Department of Statistics, UCLA, UC Los Angeles* .

R Development Core Team (2011). *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0.
URL: *http://www.R-project.org/*

R-Development-Core-Team (2012). R homepage.
URL: *http://www.r-project.org/*

Rothkopf, E. Z. (1957). A measure of stimulus similarity and errors in some pairedassociate learning, *Journal of Experimental Psychology* **53**: 94–101.

RStudio (2012). Rstudio: Integrated development environment for r. version 0.94.102.
URL: *http://www.rstudio.org/*

SAS Institute Inc. (2011). Base SAS 9.3 procedures guide. NC: SAS Institute Inc.

statsoft (2012). Statistica. the STATISTICA suite of analytics software products and solutions. **URL:** *http://www.statsoft.co.za*

Swayne, D. F., Cook, D. and Buja, A. (1998). Xgobi: Interactive data visualization in the X Window sysem, *Journal of Computational and graphical statistics* **7**: 113–130.

Swayne, D. F., Temple-Land, D., Buja, A. and Cook, D. (2002). Ggobi: Evolving from XGobi into an extensive framework for interactive data visualistion, *Journal of Computational statistical computing* .