An investigation of Multidimensional Scaling with an emphasis on the development of an R based Graphical User Interface for performing Multidimensional Scaling procedures

Author: Andrew TIMM

Supervisor: Associate Professor Sugnet LUBBE

DISSERTATION PRESENTED FOR THE DEGREE OF MASTER OF SCIENCE

IN THE DEPARTMENT OF STATISTICAL SCIENCES

UNIVERSITY OF CAPE TOWN

October 12, 2012

PUBLICATION

I hereby grant the University free license to publish this dissertation in whole or part in any format the University deems fit.

PLAGIARISM DECLARATION

1. I know that plagiarism is wrong. Plagiarism is to use anothers work and pretend that it is my own.
2. I have used the APA referencing guide for citation and referencing. Each contribution to, and quotation in this dissertation from the work(s) of other people has been contributed, and has been cited and referenced.
3. I know the meaning of plagiarism and declare that all of the work in the dissertation, save for that which is properly acknowledged, is my own.

Signature:

Date:

Acknowledgments

The author would like to thank Associate Professor Sugnet Lubbe of the University of Cape Town for constant input and supervision throughout all extents of this research. Professor Niel le Roux of Stellenbosch University for contribution of original *R* code and Professor Patrick Groenen of the Erasmus University of Rotterdam who kindly provided suggestions and necessary constructive criticism over the development of the MDS-GUI from a Multidimensional Scaling perspective.

I would also like to thank my parents, Jill and Tony, for their support. And finally Catherine Stephenson, for her caring and daily encouragement.

Abstract

This dissertation is centered around the development of a graphical user interface, using the *R* statistical programing language, for performing Multidimensional Scaling. This program is called the MDS-GUI. Multidimensional Scaling (MDS) is one of the groups of multivariate analysis techniques that is used for dimension reduction. In general, these methods of MDS can be viewed as the problem of constructing a map when given a set of interpoint distances. The graphical configuration is produced, usually in two or three dimensions, in such a way that objects of the data are represented by points, where the Euclidean distances between them optimally represents the given set of observed distances.

The MDS-GUI was developed using a combination of *R* and the scripting language *tcltk*. The primary objective of its design was to provide a comprehensive range of MDS methods and analytical tools that are accessed in a point and click manner. The target user group of the software is therefore widely spread as no coding and only a little expertise on MDS is required for its use.

The capabilities of the MDS-GUI are demonstrated with the use of three data sets. The first is the well known and well used Morse-Code data; the second is a synthetic microarray based data set; and the third concerns the nutritional contents of a group of the cereals from the Kellog's company.

The program will be the first complete MDS based GUI for the *R*-Environment, and will also be the package that provides access to the widest range of MDS methods in *R*.

Contents

1 Introduction **1**

1.1 Introduction 1

1.2 Background Information 2

1.3 Objectives 3

1.4 Scope and Limitations 3

1.5 Layout of Document 4

1.6 Notation 5

2 The Theory of Multidimensional Scaling **7**

2.1 An Introduction to Ordination 7

2.2 Multidimensional Scaling 8

2.2.1 Metric Multidimensional Scaling 11

2.2.2 Non-Metric Multidimensional Scaling 11

2.3 The General MDS Algorithm 12

2.4 Stress & Strain 16

2.4.1 General Form of Stress 16

2.4.2 STRESS-1 17

2.4.3 STRESS-2 18

2.4.4 Normalised Raw Stress 18

2.4.5 Strain 19

2.4.6 Interpretation of Stress 19

2.5 Dimensionality 21

2.5.1 Euclidean Embedding and Dimensions 23

2.6 Diagnostic Tools 23

2.6.1 Scree Plot 24

2.6.2 Shepard Plot 26
2.7 Data 30
2.7.1 Types of Data 30
2.7.1.1 Nominal Scale 31
2.7.1.2 Ordinal Scale 31
2.7.1.3 Interval Scale 31
2.7.1.4 Ratio Scale 31
2.7.2 Modes and Ways 31
2.7.3 Transformations of Data 32
2.8 Measures of Proximity 32
2.8.1 Measuring Dissimilarities 33
2.8.1.1 Euclidean Distance 33
2.8.1.2 Weighted Euclidean Distance 34
2.8.1.3 Mahalanobis Distance 34
2.8.1.4 City Block Metric 34
2.8.1.5 Minkowski Metric 35
2.8.1.6 Canberra Metric 36
2.8.1.7 Bray-Curtis Distance 36
2.8.1.8 Soergel Distance 36
2.8.1.9 Bhattacharyya Distance 37
2.8.1.10 Wave-Hedges Distance 37
2.8.1.11 Angular Separation 37
2.8.1.12 Divergence 38
2.8.2 Measuring Similarities 38
2.8.2.1 Jaccard Coefficient 40
2.8.2.2 Simple Matching Coefficient 40
2.8.2.3 Correlation 41
2.9 Setting up of MDS Procedure 42
2.10 Interpretation of MDS Results 45
2.10.1 Interpreting Configuration 45
2.10.2 Interpreting Axes and Dimension 51
2.10.3 Comparing Configurations 52
2.11 Applications for MDS 55
2.12 Multivariate Methods Related to MDS 56

2.12.1 Principal Component Analysis 56
2.12.2 Correspondence Analysis 57
2.12.3 Cluster Analysis 57
2.13 Drawbacks of MDS 58

3 Mathematics of Multidimensional Scaling 60
3.1 General Mathematical Results 60
3.1.1 Differentiation of Euclidean Distances 60
3.2 Metric MDS 61
3.2.1 Classical Scaling 62
3.2.1.1 Classical Scaling using Optimisation 64
3.2.2 Least Squares Scaling 65
3.3 Non-Metric MDS 68
3.3.1 Kruskals MDS 70
3.3.2 Sammon Mapping 72
3.4 SMACOF 73
3.4.1 The Majorisation Algorithm 73
3.4.2 Metric SMACOF 75
3.4.3 Non-Metric SMACOF 77
3.5 Unidimensional Scaling 77
3.6 INDSCAL 78

4 Technical Components 84
4.1 Tcl & tcltk 84
4.1.1 Integrating Languages 85
4.1.2 tcltk 85
4.1.3 Features and Benefits of tcltk 86
4.2 R 87
4.2.1 The *R*-Environment 87
4.2.2 *R*-Packages 88
4.2.2.1 stats package 89
4.2.2.2 MASS package 89
4.2.2.3 tcltk package 91
4.2.2.4 tkrplot package 92

4.2.2.5 rpanel package 93
4.2.2.6 tcltk2 package 93
4.2.2.7 RColorBrewer package 93
4.2.2.8 boot package 94
4.2.2.9 RGL package 95
4.2.2.10 scatterplot3d package 95
4.2.3 Other R-Packages of Interest 95
4.2.3.1 smacof package 96
4.2.3.2 homals package 96
4.2.3.3 SensoMineR package 97
4.2.3.4 Limma package 97
4.2.3.5 Labdsv package 98
4.2.3.6 vegan package 98

5 The MDS-GUI 99

5.1 An Introduction to the MDS-GUI 99
5.1.1 A Tour 100
5.2 Multidimensional Scaling Capabilities 102
5.3 Menu Structures of the Software 102
5.3.1 Top Menu 102
5.3.1.1 File Menu 103
5.3.1.2 General Menu 103
5.3.1.3 Data Menu 104
5.3.1.4 Multivariate Tools Menu 106
5.3.1.5 Help Menu 108
5.3.2 Main Plot Menu 110
5.3.3 Secondary Plot Menus 111
5.3.4 General Settings Menu 111
5.3.4.1 General Tab (a) 111
5.3.4.2 Convergence Tab (b) 112
5.3.4.3 Graphical Tab (c) 113
5.3.4.4 Visualisaiton Tab (d) 113
5.3.5 Data Options Menu 113
5.3.6 MDS Options Menu 114

5.3.6.1 Dimensions Tab 114
5.3.6.2 Starting Configurations Tab 115
5.3.6.3 Stress Tab 116
5.3.7 Plot Option Menus 116
5.3.7.1 General Tab 116
5.3.7.2 Points Tab 118
5.3.7.3 Lines Tab 118
5.3.7.4 Axes Tab 118
5.4 Overview of Features 118
5.4.1 Plotting Tab Features 119
5.4.2 $p \geq 2$ 120
5.4.2.1 Three Dimensions 120
5.4.2.2 More Than Three Dimensions 122
5.4.3 Shepard Plot 123
5.4.3.1 Shepard Point Labeling 124
5.4.3.2 Brushing the Shepard Plot 125
5.4.4 Scree Plot 128
5.4.5 Iterations Observable 130
5.4.5.1 Stress Plots 130
5.4.5.2 Progress Bar and End Process 131
5.4.6 Procrustes Analysis 132
5.4.7 Point Labeling 133
5.4.8 Manual Alterations of Configurations 135
5.4.9 Zoom 137
5.4.10 Configuration Orientation 138
5.4.11 Colour Options 140
5.4.12 Variable Axes 144
5.4.13 Removed Points and Axes 145
5.4.14 Notes/Script 147
5.4.15 Save and Load Workspace 148
5.4.16 Export to PDF 149
5.4.17 Print 151
5.4.18 Supporting Plotting Features 152
5.5 Coding 153

5.5.1 MDS-GUI Development 153
5.5.2 Challenges 154
5.6 The MDS-GUI: Version 2 154
5.7 Similar Software 155
5.7.1 iMDS 156
5.7.2 XGVis and GGVis 156
5.8 The MDSGUI Package and Supporting Documentation 157
5.8.1 User Manual 157
5.8.2 **MDSGUI** Package Reference Manual 158
5.8.3 Vignette 158

6 Application of the MDS-GUI 159

6.1 Morse-Code Data 159
6.1.1 Morse-Code: General Analysis 160
6.1.2 Morse-Code: Configuration Analysis 163
6.2 SynTReN Microarray Data 168
6.2.1 SynTReN Microarray: General Analysis 170
6.2.2 SynTReN Microarray: Configuration Analysis 173
6.3 Breakfast Cereal Data 177
6.3.1 Cereal Data: General Analysis 179
6.3.2 Cereal Data: Configuration Analysis 183

7 Conclusions & Recommendations 188

7.1 Significance of Research and Development 188
7.2 Concluding Remarks About Objectives 189
7.3 Data Study Conclusions 191
7.4 Recommendations 192

A Data Sets A.1-1

A.1 Skulls Data A.1-1
A.2 Morse-Code Data A.2-1
A.3 Breakfast Cereal Data A.3-1
A.4 SynTReN EColi Microarray Data A.4-1

B Supporting Documentation B.1-1

B.1 Reference Manual B.1-1

B.2 Users Manual B.2-1

B.3 Vignette B.3-1

List of Figures

2.1 Example of MDS Ordination Configuration 10
2.2 Accuracy of Distance 21
2.3 Scree Plot Example 24
2.4 Scree Slope 25
2.5 Shepard Diagram Examples 28
2.6 Example of Shepard Plots With One Distorted Point 29
2.7 Skull Data Example 47
2.8 Skull Data With Coloured Categories 49
2.9 Skull Data: Cluster 50
2.10 Skull Data: Axes of Variation 53
2.11 Skull Data: Procrustes Analysis Example 55

3.1 Non-Metric MDS: Transformation of Distances 69
3.2 Majorising Algorithm Example 74

5.1 The MDS-GUI 101
5.2 The File Top-Menu 103
5.3 The General Top-Menus 104
5.4 Appearance Settings 104
5.5 The Data Top-Menus 105
5.6 Uploaded Data Menu 106
5.7 The Multivariate Tools Top-Menus 107
5.8 The Help Top-Menu 108
5.9 Function Help 109
5.10 About: Text Box 110
5.11 Main Plot Options Menu 110

5.12 Secondary Plot Options Menu 111
5.13 General Settings Menu 112
5.14 Data Options Menu 114
5.15 MDS Options Menu 115
5.16 *p=1* Warning 116
5.17 Plot Options Menu 117
5.18 Configuration Table 119
5.19 Three Dimensions Options 120
5.20 3D Plotting 121
5.21 Large Dimensions Options 123
5.22 Matrix Editor 124
5.23 MDS-GUI: Displaying Shepard Plot 125
5.24 Labeled Shepard Points 126
5.25 MDS-GUI: Specific Shepard Point Label 127
5.26 Shepard Point Brushing 127
5.27 Scree Plot 129
5.28 Stress Plots 132
5.29 Progress Bar and End Process Button 133
5.30 Procrustes Analysis Example 134
5.31 Label Point with Cursor 135
5.32 Label Specific Point 135
5.33 Moving Configuration Points 137
5.34 Manual Zoom Controls 138
5.35 Zoom Options 139
5.36 Configuration Orientation Controls 140
5.37 Configuration Orientation Changes 141
5.38 Colour Categories 142
5.39 Colour Tools 143
5.40 Change Point Colour 144
5.41 Display Variable Axes 146
5.42 Variable Axes Removal 146
5.43 Removed Item Tables 147
5.44 Notes-Script Tab 148
5.45 Export Instruction Message 149

5.46 PDF Output: Plot1-Page1 150
5.47 PDF Output: Plot1-Page2 151
5.48 Popped-Out Plot 152

6.1 Morse-Code Data: Best Results 162
6.2 Morse Code: Procrustes Analysis 163
6.3 Morse-Code Data: Diagnostic Plots 164
6.4 Morse Code: Symbol Categories 165
6.5 Morse Code: Length Categories 166
6.6 Morse-Code Data: Labeled Points 167
6.7 True Gene Association Network 170
6.8 MicArray: Procrustes Analysis 172
6.9 MicArray: Metric SMACOF (City-Block Metric) 173
6.10 MicArray: Diagnostic Plots 174
6.11 MicArray: Metric SMACOF with Coloured Groups 175
6.12 MicArray: Deviating Pairs 177
6.13 MicArray: p=3 Configuration 178
6.14 MicArray: p=3 Shepard Plot 179
6.15 Cereal Data: Shepard Plot Comparison 181
6.16 Cereal Data: Kruskal's Analysis Scree Plot 182
6.17 Cereal Data: Kruskal's Analysis Configuration 183
6.18 Cereal Data: Kruskal's Analysis Configuration (Axes) 184
6.19 Cereal Data: Kruskal's Analysis Configuration (Shelf Axes) . 186
6.20 Cereal Data: Furthest Points 187

List of Tables

2.1 Distance Matrix Example 9

2.2 Coordinate Vectors: $p = 2$ 9

2.3 Interpretation of STRESS-1 19

2.4 Binary Data Similarity Coefficient Key 39

6.1 MDS Stress Values on Morse Code Data 160

6.2 NRS Values on Microarray Data 170

6.3 Stress-2 Values on Cereal Data 180

A.1 Skull Variables A.1-2

A.2 Skulls Data A.1-2

A.3 Morse Code Symbols A.2-1

A.4 Asymmetric Morse-Code Data A.2-2

A.5 Symmetric Morse-Code Data A.2-4

A.6 Kellogg's Breakfast Cereal A.3-1

A.7 Breakfast Cereal Variables A.3-2

A.8 Kellog's Cereal Data A.3-2

A.9 Microarray Ecoli Genes A.4-1

Chapter 1

Introduction

1.1 Introduction

The use of statistical methods in the field of data analysis has expanded beyond being strictly performed by statisticians. Areas of research, including Biology, Sociology, Psychology, Marketing, Genealogy and Ecology, among others, are making increasing use of advanced statistical computational methods in their research. With this ever growing demand for these procedures comes the necessity for statisticians to produce means of providing these researchers with the ability to undertake such endeavors. Statistical programming languages, such as *R* and *Stata*, and more commercially, the *SAS* software and *STATISTICA*, are available to statisticians and non-statisticians alike. These programs however require a certain amount of prior statistical knowledge and have a steep learning curve. This is something that many non-statisticians may find intimidating.

An emerging niche among statistical programmers has become prevalent over recent years, and this is in the development of Graphical User Interfaces (GUIs) that allow for easy implementation of specific complicated statistical functions. The subject of this dissertation is based on Multidimensional Scaling (MDS) and the development of the MDS-GUI, which stands for the Multidimensional Scaling Graphical User Interface. This piece of software is designed for the *R* programming language and supplies the user with an interface that is easily navigable and is capable of performing and analysing multiple methods of MDS.

1.2 Background Information

These statistical GUIs are developed with the primary intention of ease of use and interactability. As such, the programs are usually designed such that they are navigated and utilised in a point and click manner. The key benefit of such programs is that the coding aspect is removed from the process and they are therefore accessible by those who are either incapable of such computer programming, or not knowledgeable enough about the statistics of the functions to perform the task themselves. This trend of GUI development is particularly prevalent within the *R* programming community, with such examples as the BiplotGUI (la Grange et al., 2009) and the caGUI (Markos, 2010) which were designed to simply and efficiently perform biplot and correspondence analysis functions respectively. *R* is a convenient environment in which to create these types of software as it is free to all users and driven by user contribution of packages. A number of these packages are geared to aid with GUI development, such as the `tcltk` package (R Development Core Team, 2012) which provides *R* functionality of the *tcltk* development language. A notable gap in the *R* database was found to be that there was a lack of GUI for performing Multidimensional Scaling techniques, which are highly useful in multivariate data analysis. This provided the opportunity to develop a suitable and interactive piece of software that provided all *R* users with the ability to perform MDS operations easily and without having to undertake the coding themselves.

The ordination methods of Multidimensional Scaling have been studied and used for decades and numerous works have been written on the subject. The two primary sources of information for the MDS topics covered in this dissertation were *Multidimensional Scaling: Second Edition* (Cox and Cox, 2001) and *Modern Multidimensional Scaling Theory and Applications: Second Edition* (Borg and Groenen, 2005).

1.3 Objectives

The objectives of the dissertation are related to both investigating MDS techniques and development of the MDS software. These objectives are summarised by the following:

1. Provide suitable information regarding Multidimensional Scaling and its methods in the context of the MDS-GUI. This includes full theoretical and mathematical explanations of the MDS algorithms, diagnostic tools and interpretation of its results.
2. Provide Information regarding the programming languages and packages used in the development of the MDS-GUI.
3. Develop a fully functional version of the MDS-GUI. The software, upon completion, will become available for download from the relevant *R* websites.
4. Describe the MDS-GUI in full, including all components of the interface and complete discussion of its functionality.
5. Simultaneously demonstrate the results of the MDS-GUI and provide examples of interpretation of MDS based results.

1.4 Scope and Limitations

Multidimensional Scaling contains a broad number of topics and procedural aspects. The concepts behind many of these components of MDS theory are themselves vast and have been considered as research topics alone. As a result, most of these will not be investigated and will be taken as given during the presentation of theoretical concepts. An example of one such aspect which will not be investigated is the optimisation methods used in finding optimal results. Another feature of MDS is the vast number of methods that fall under the MDS category. The MDS-GUI only incorporates a subset of these, meaning that some MDS methods will not be featured in either the software or the documentation. The MDS-GUI will provide means of performing Classical Scaling, Least Squares Scaling, Metric and Non-Metric

SMACOF, Kruskal's Analysis and Sammon Mapping. The statistical scope of the dissertation includes these six methods.

The scope, from a coding point of view, specifically involves the development of an interface for performing MDS procedures. The backing code however made use of preexisting code written to retrieve MDS results. This code was supplied by associated contributors and existing *R* packages.

This document, although addressing the development of the MDS-GUI to great extent, will not provide any *R* or *tcltk* code during the discussions. Accompanying the document will be a disk containing all pieces of code related to the MDS-GUI (and the GUI itself). Interested readers are invited to look at this code for a better idea of the coding processes involved in the GUI functions.

1.5 Layout of Document

The layout of the remainder of this dissertation aims to first provide sufficient information regarding Multidimensional Scaling and then discuss the MDS-GUI itself. The remaining six chapters will cover the following.

Chapter 2 provides information regarding the theoretical concepts of Multidimensional Scaling. The Chapter discusses MDS and its results in general terms and without making too much distinction between the various methods of MDS. This discussion includes an analysis of the MDS algorithm, stress and what a researcher is required to do to perform MDS when analysing their data.

Chapter 3 is focused on the mathematics of MDS. This includes descriptions of Metric and Non-Metric methods, isotonic regression, the SMACOF algorithm and the six MDS methods incorporated into the MDS-GUI.

Chapter 4 discusses the technical computer related aspects of the project. It provides information on all coding languages and packages that are utilised throughout the development of the MDS-GUI software.

Chapter 5 introduces the MDS-GUI and describes it in detail. The Chapter firstly gives a tour of the interface by describing each of the areas of the front-end of the software and the menus that control it. Following this, detailed descriptions and demonstrations of all the major functions of the MDS-GUI are given. The Chapter ends off with short sections on the challenges of development, similar software and the `MDSGUI` *R* package.

The MDS-GUI is then put to use in Chapter 6, where its features and output are demonstrated with the use of three different sets of data. These data sets were selected to highlight the varying features that are applicable to different types of data. Interpretations of the results from a statistical point of view are also provided.

The document ends with Chapter 7, giving the conclusions and recommendations that were accumulated throughout the study and development process.

1.6 Notation

For convenience of the reader, the notation used throughout the document will be summarised here. Each example will be introduced in detail in the text. This Section simply provides a means of reference. In general, all matrices are referenced by an upper case boldface letter and elements of the matrix by the appropriate lower case letter.

- n: Number of objects/subjects in the data.
- m: Number of variables of the data.
- p: number of MDS plotting dimensions.
- $\mathbf{Z}$: Data matrix in the form objects$\times$variables, i.e. with dimensions $n \times m$.
- $\mathbf{\Delta}$: Symmetric dissimilarity matrix of objects. Dimensions are $n \times n$.
- δ_{ij}: Observed dissimilarity between the i^{th} and j^{th} objects.
- $\mathbf{S}$: Similarity matrix of objects. Dimensions are $n \times n$.
- s_{ij}: Observed similarity been the i^{th} and j^{th} objects.
- $\mathbf{X}$: MDS Coordinate matrix. Dimensions are $n \times p$.
- $\mathbf{D}$: Symmetric Matrix of Euclidean distances between points in $\mathbf{X}$.
- d_{ij}: Euclidean distance between the i^{th} and j^{th} objects in $\mathbf{X}$.

- $\hat{\mathbf{D}}$: Matrix of disparities. Derived from admissible transformation of proximities.
- $\hat{d}_{ij}$: Disparity between the i^{th} and j^{th} object.

The following convention will be followed with respect to programming languages, *R* packages and functions.

- Programming languages will be italisised, e.g. *R*.
- *R* packages will be in boldface, e.g. **MASS**.
- Functions will be written in a typewriter-style font, e.g. `sammon`.

Chapter 2

The Theory of Multidimensional Scaling

This Chapter serves to inform the reader on the theoretical concepts of Multidimensional Scaling with a focus more on the procedural aspects of the topic rather than the in-depth Matrix Algebra. This Mathematical theory will be covered in the following Chapter *The Mathematics of Multidimensional Scaling*.

The content of the Chapter will include the theory of MDS as a process. It will also describe the diagnostic tools used to analyze the performance of the methods, as well as the place that Multidimensional Scaling holds in practical data analysis.

2.1 An Introduction to Ordination

Ordination is a general term for techniques in multivariate analysis which adapts sizable matrices of data in such a way that when projected onto a space of fewer dimensions, intrinsic patterns within the data may be visually inspected (Clark, 2005). There are a number of ordination methods that have been developed, including: Factor Analysis, Correspondence Analysis, and Principle Component Analysis. The subject of this dissertation, however, is the group of ordination methods collectively known as Multidimensional Scaling. Some texts refer to Multidimensional Scaling as *Princi-*

pal Coordinate Analysis, however a more accurate description would be that Principal Coordinate Analysis is an alternative name to what is called "Classical Scaling", which is only one of the many forms of MDS. Each of these methods of ordination, while conceptually producing similar visual outputs, are vastly different and have their own specific uses. A more detailed view on the primary similarities and differences between Multidimensional Scaling and the other methods of ordination can be found in Section 2.12 of this Chapter under the heading *Multivariate Methods Related to MDS*.

2.2 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 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*. Stress and the other criterion measures for goodness-of-fit will be analysed later in this Chapter in Section 2.4 under the heading *Stress & Strain*. 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 de Leeuw (2008) describe MDS as a set of methods for discovering "hidden structures in multidimensional data". As the procedure only tries to match the mapped, Euclidean distances, as closely as possible to the matrix of proximities, it is not so much an exact procedure as rather a way to rearrange the points (objects) in an efficient manner so as to arrive at the configuration that best approximates the observed distances. The following example is trivial, and merely serves to demonstrate the relationship between proximity data and visual mapping. Table 2.1 demonstrates the distances, measured with an Euclidean metric, between five points. These points have been labeled from *A, B, C, D, E.* Since the proximities in this case are specifically distance based, the proximity type is clearly dissimilarity. A Classical Scaling procedure with p equal to two, when performed with this data as an input, produces the graphical results shown in Figure 2.1 with corresponding vectors in Table 2.2.

Table 2.1: Distance Matrix Example

	A	**B**	**C**	**D**	**E**
A	0	3	3	4.24	0.5
B	3	0	4.24	3	2.77
C	3	4.24	0	3	2.77
D	4.24	3	3	0	3.74
E	0.5	2.77	2.77	3.74	0

Table 2.2: Coordinate Vectors: $p = 2$

	V1	**V2**
A	-1.802	0.000
B	0.346	2.120
C	0.346	-2.120
D	2.429	0.000
E	-1.285	0.011

The MDS procedure reveals that the data describes a perfect square with

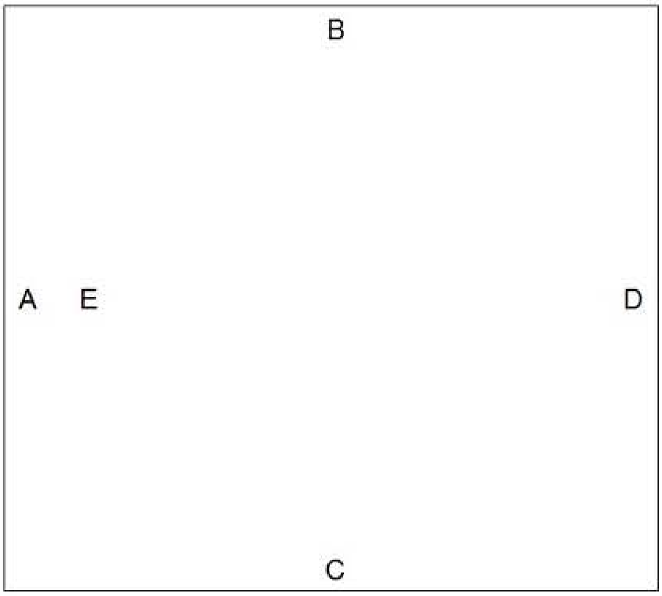


Figure 2.1: Example of MDS Ordination Configuration

points *A, B, C, D*. The fifth object *E*, is shown to be a point within the square with a strong relationship to object *A*. This example is unfortunately unrealistic in that the MDS representation of the proximities is just about perfect. Real applications are expected to have a certain amount of distortion in their results. It should be noted that the plot itself has an aspect ratio of one. A square grid is very important when producing MDS results as it preserves equality in the ratio of distances over the plotting dimensions.

It is the norm that the p-dimensional space onto which these points are mapped is Euclidean. While this is not theoretically a requirement, it is the only logical way of portraying the solutions as MDS aims to take complicated information and portray it in a format accessible to a human observer. The proximity matrix however is often found to be calculated using distance measures other than Euclidean. A comprehensive list of methods for distance measurement is provided and discussed in Section 2.8. The number of dimensions onto which the points are mapped is also typically either one, two or three as these are the only options for producing visual images comprehensible to the human brain. The MDS procedures are, however, in no way limited in terms of p. The number of dimensions that are used in the procedure allow us to explain proximities of the data in that number of dimensions. In an example of distances between cities, one could explain distances in terms of two dimensions, being north/south and west/east. Alternatively a further dimension could be included which could account for the altitude of the cities.

While there are a wide range of types of Multidimensional Scaling, there are two categories under which the various methods fall. These two categories are Metric and Non-Metric Multidimensional Scaling, both of which will be discussed in brief detail here.

2.2.1 Metric Multidimensional Scaling

The techniques of Metric Multidimensional Scaling are structured in such a way that there is an assumption of metric qualities in the measurement of the proximities. That is to say that the extent of the (dis)similarities between points are taken into account. Thus the distances in metric MDS space preserve the intervals and ratios as well as possible (Wickelmaier, 2003). The use of Metric Multidimensional Scaling is only valid when the assumption of metric distances can be justified. Methods that fall under Metric Multidimensional Scaling include: Classical Scaling (Principal Coordinate Analysis), Least Squares Scaling and Metric SMACOF. All Metric MDS methods are discussed in detail in Section 3.2.

2.2.2 Non-Metric Multidimensional Scaling

In many situations the metric assumptions described above are too strong for the data at hand. It is under these circumstances that the use of Non-Metric Multidimensional Scaling techniques may be more appropriate. Under the theories of non-metric multidimensional scaling the extent of the proximities is irrelevant. Only the ordering of the (dis)similarities is factored during the derivation of the MDS configuration. Methods that fall under Non-Metric Multidimensional Scaling include: Kruskal's MDS, Non-Metric SMACOF and Sammon Mapping. All Non-Metric MDS methods are discussed in detail in Section 3.3.

2.3 The General MDS Algorithm

Within Multidimensional Scaling there are a number of different subsets of MDS techniques, most with a classification of either metric or non-metric. A number of these methods will be described and investigated in detail in Chapter 3 *The Mathematics of Multidimensional Scaling.* The majority of the methods, however, follow the same general algorithm in terms of how the procedure progresses and at what point it terminates. This basic algorithm is described below.

The process of Multidimensional Scaling may begin in one of two ways. The first of these is with a data matrix $\mathbf{Z}$, consisting of n rows of samples and m columns of variables. The $n \times n$ proximity matrix used in the MDS process is derived from this data matrix. As mentioned before, this proximity matrix can be in the form of dissimilarities, $\mathbf{\Delta}$, or similarities, $\mathbf{S}$, between the samples. Alternatively, the $n \times n$ proximity matrix is provided independently and in this case, no proximity calculation is required. A typical method of dissimilarity calculation is using an Euclidean measure to find the pairwise distances between samples. The matrix does not necessarily have to be symmetric as, depending on how the proximity values are calculated, the differences between two objects might be different depending from which direction the measure was taken. For example, if elevation is being assessed the difference between two points would yield both a positive and negative value depending on the starting point. This proximity matrix is however required to be square and should, as best as possible, be complete. Once the proximity matrix has been derived the data collection is complete. Many MDS algorithms specifically require the dissimilarity matrix, $\mathbf{\Delta}$, as input. In this case similarities are simply transformed to dissimilarities. The elements of the dissimilarity matrix are δ_{rs} and refer to the original measured proximity between the r^{th} and s^{th} objects. Some MDS methods and programs make allowances for asymmetry in their proximity data. The software discussed in this dissertation, however, strictly requires all proximities to be symmetric. The following criteria therefore must hold.

$$\delta_{ij} = \delta_{ji}$$

$$\delta_{ij} = 0 \quad iff \quad i = j$$

$$\delta_{ij} + \delta_{jk} \leq \delta ik \quad \forall i, j, k$$

Before the MDS procedure commences, the desired number of dimensions must be chosen for the ordination. Further in this Chapter the concept of Scree Plots will be discussed. These plots aid the researcher in determining the most appropriate number of p dimensions that should be used for the particular data. However, for all intents and purposes of describing this simple algorithm, the specific number of dimensions being used is irrelevant so no further detail on the dimensionality of the ordination need be given at this point. With the proximity matrix calculated and p, the number of dimensions, decided upon, the Multidimensional Scaling procedure may commence. The process begins with an initial configuration of points in the p dimensional space, with each point representing an object from the data. This configuration, described by the coordinate matrix $\mathbf{X}$ with dimensions $n \times p$, can be entirely random or else could be based on some other prior knowledge, such as the results of some other ordination, or a pattern suspected by the researcher. It should be noted however that due to the problem of local minima, the initial configuration may influence the final result. This problem of local minima will be discussed in Section 2.13. The distances between points in the configuration are then calculated. This calculation is almost exclusively done using an Euclidean Metric as the plotting plane that is observed by the researcher is Euclidean by nature. This symmetric matrix of ordination based distances has dimensions $n \times n$ and is referred to as $\mathbf{D}$ (d_{rs} is the distance between the r^{th} and s^{th} objects within the MDS configuration). In the cases of Non-Metric Scaling, these ordination based distances are then regressed against the original distance matrix $\boldsymbol{\Delta}$ using one of a number of regression methods, and are fitted using a method of least squares. The transformed distances are generally referred to as $\hat{d}$ distances, and form part of the symmetric $n \times n$ matrix $\hat{\mathbf{D}}$. All fitted elements of $\hat{\mathbf{D}}$ are referred to as $\hat{d}_{rs}$. Metric cases see the method of least squares fitting simply the ordination distances and the proximities.

$$\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} (f(\delta_{ij}) - d_{ij})^2 = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} (\hat{d}_{ij} - d_{ij})^2 \tag{2.1}$$

The sum of squares component in (2.1) is the basis of the function usually referred to as stress, which is an integral component of most Multidimensional Scaling techniques. Stress is the general term and actually comes in a number of different forms, the most important of which are described in the following Section. The common interpretation of stress is however, that the smaller the stress the better the fit, as there is a smaller (squared) difference between the distances in the configuration of points and the original proximities.

With stress as an indicator of goodness-of-fit of a configuration, the MDS process begins to move the points of the configuration. The nature of how the points are moved is dependent on the method of MDS being used. After each adjustment, the matrix of ordination distances, $\mathbf{D}$, is calculated (and in Non-Metric cases, so is $\hat{\mathbf{D}}$) and subsequently so is the stress of the configuration. If a new configuration yields a stress value greater than the configuration before it, it is discarded. Alternatively, when a new configuration yields a smaller stress value than before, the old configuration is discarded and the new configuration becomes active. Many formats of MDS have only monotonically decreasing stress, meaning that any stress value will only ever be smaller than or equal to its preceding value. The configuration is thus improved by moving the points of the ordination in small amounts in the direction that causes the stress to decrease most rapidly. This procedure is then continued until either the stress has reached a value small enough to satisfy the researcher (a certain threshold value has been achieved) or a point of convergence has been reached. Convergence is reached when the difference in stress between two consecutive configurations is smaller than a certain tolerance that has been predefined. This implies that a larger predefined tolerance will usually cause convergence sooner than when a smaller tolerance is used. None the less, this point of convergence indicates that the accuracy of the configuration is unable to improve and thus a minimum stress value (local or global) has been met.

At this point the procedure has stopped. In the event that convergence occurs at a point where stress is considered unacceptably high, the researcher must make the choice of either making adjustments to the setup of the procedure (starting configuration, tolerance, etc.) or abandoning MDS in favour of a more appropriate method of ordination for their specific needs. Another option that may be available is to increase p, the number of dimensions of the ordination configuration. As will be explained in Section 2.5, increasing p will always decrease the final stress value and thus demonstrate a better fit. On the other hand, in the event that the process stopped due to stress reaching an acceptably low value, the researcher is likely to be satisfied with the outcome of the MDS procedure. The resulting configuration, with coordinates $\mathbf{X}$, will thus display the arrangement of the points that best represents the proximity matrix in the chosen p dimensions. The actual orientation of the axes in this final configuration solution is arbitrary. The decision of how the ordination should be oriented is usually a subjective decision of the researcher, who is likely to make a decision based on what is considered the most easily interpretable orientation. For example, North-South is likely to be appropriate as the vertical on a two dimensional geographical point map. The rotation of the configuration is irrelevant in the context of MDS since distance is indifferent under rotation. The MDS result is thus unlikely to have North-South on the vertical, which leaves the final rotation at liberty of the researcher.

The above procedure can be summarized in the following steps.

1. Assign objects to points in initial configuration in the p dimensional space. The $n \times p$ coordinate matrix is referred to as $\mathbf{X}$.

2. Compute distances among all pairs of points in configuration to form a matrix of distances, $\mathbf{D}$.

3. When performing Metric MDS, do 3(a). When performing Non-Metric MDS, do 3(b).

   (a) The $\mathbf{D}$ matrix is then compared to the original matrix of proximities, $\mathbf{\Delta}$. Stress is then calculated.

   (b) The $\mathbf{D}$ matrix is then compared to $\hat{\mathbf{D}}$, the fitted values resulting from some function of $\mathbf{\Delta}$. Stress is then calculated.

4. The coordinates, $\mathbf{X}$, are adjusted in such a way that stress is improved upon from the previous configuration.

5. Steps two to four are repeated until either:
   (a) Stress has reached an acceptably low point, or
   (b) The value of stress has converged given the set tolerance.
   (c) The value of stress has not converged given the set tolerance, but the maximum number of iterations have been reached

In the case of 5(b) or (c) where stress is unacceptably high, do 6.

6. Perform one or more of the following.
   (a) Tolerance value decreased.
   (b) Starting configuration altered.
   (c) p is increased.
   (d) Or alternatively, MDS is abandoned.

2.4 Stress & Strain

The previous Section made extensive reference to the functions collectively known as "stress" in the computation procedure during the Multidimensional Scaling algorithm. While there do exist other measures to evaluate the degree of correspondence between the distance among points implied by the MDS map and the observed proximity matrix, the various forms of stress have become the norm within Multidimensional Scaling.

2.4.1 General Form of Stress

The general form of the various stress functions is as follows:

$$\sqrt{\frac{\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}(f(\delta_{ij})-d_{ij})^2}{scale}} \tag{2.2}$$

In the equation δ_{ij} refers to the ij^{th} element of the $n \times n$ observed proximity matrix $\mathbf{\Delta}$, and $f(\delta_{ij})$ is some smoothing function of this data (Cox and

Cox, 2001). The d_{ij}, as described, are the elements of the square matrix, $\mathbf{D}$, of distances between points across all dimensions in the resulting MDS configuration. $\mathbf{D}$ is assumed to be symmetric, thus $d_{ij} = d_{ji}$, $d_{ii} = 0$, $i = 1, ..., n-1$ and $j = i+1, ..., n$. Finally, the 'scale' component refers to a constant scaling factor, used to keep the value of stress within the convenient range of between 0 and 1. Stress therefore acts as an inverse measure of the goodness-of-fit of an MDS configuration, as stress closer to 0 indicates a good fit (with stress of zero indicating a perfect mapping of the points) and stress values closer to 1 indicating a very poor fit.

The transformations of the input values $f(\delta_{ij})$ used depends on the type of MDS that is being used to perform the operation. Some appropriate and relevant transformations will be observed and discussed in both Section 2.7, the *Data* Section of this Chapter, and throughout Chapter 3, but briefly the two major cases will be mentioned here. In metric scaling $f(\delta_{ij}) = \delta_{ij}$ (the identity transformation), which means that the raw input proximity data is compared directly to the mapped distances. In non-metric scaling however, $f(\delta_{ij})$ is usually in the form of a monotonic transformation of the MDS proximities, and is used to minimise the stress function.

The general form of the stress equation, is given by (2.2). The three most common forms of stress will now be discussed, these being: STRESS-1 (or Kruskal's Stress); STRESS-2; and Normalized Raw Stress.

2.4.2 STRESS-1

The most common form of stress, at least within non-metric MDS, was developed by (Kruskal, 1964) and is known as *Kruskal's Stress*, or more commonly, *STRESS-1*. Throughout this dissertation it will be referred to exclusively as STRESS-1. In this version of stress, the scaling factor is $\sum\sum d_{ij}^2$ which is the sum of squared distances between all pairs of points comprising the MDS mapped configuration of points. The resulting complete formula for STRESS-1 is:

$$STRESS1 = \sqrt{\frac{\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}(f(\delta_{ij}) - d_{ij})^2}{\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} d_{ij}^2}} \tag{2.3}$$

2.4.3 STRESS-2

STRESS-2 is an alternative to STRESS-1 and differs only with regards to the scaling factor in its calculation. The scaling factor becomes $\sum\sum(d_{ij} - d..)^2$ where $d..$ is the overall mean distance of all pairs of points combined in the ordination configuration. The equation 2.3 shows the complete formula for STRESS-2.

$$STRESS2 = \sqrt{\frac{\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}(f(\delta_{ij}) - d_{ij})^2}{\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}(d_{ij} - d..)^2}} \tag{2.4}$$

The scaling factor of STRESS-2 places more of a restriction on the configuration and results in higher stress values (Cox and Cox, 2001) and for this reason is often considered less desirable than STRESS-1.

2.4.4 Normalised Raw Stress

Normalised Raw Stress (NRS) is another form of Stress that is widely used. It is defined as the squared version of STRESS-1 and thus has the following formula:

$$NRS = \frac{\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}(f(\delta_{ij}) - d_{ij})^2}{\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} d_{ij}^2} \tag{2.5}$$

The effect of squaring the STRESS-1 value results in normalised raw stress scores being much lower than those of STRESS-1. This version of stress can thus be considered to be less strict than others.

2.4.5 Strain

STRAIN is a loss function that is specifically applicable to Classical Scaling. While the stress values already discussed may be used on Classical Scaling examples with the same interpretation as other MDS methods, the option of Strain is also applicable. The STRAIN formula is given by equation (2.6), where the $n \times m$ data matrix is given by $\mathbf{Z}$ and the matrix of MDS configuration coordinates is given by $\mathbf{X}$.

$$STRAIN = tr[(\mathbf{Z}\mathbf{Z}^T - \mathbf{X}\mathbf{X}^T)^T(\mathbf{Z}\mathbf{Z}^T - \mathbf{X}\mathbf{X}^T)] \tag{2.6}$$

2.4.6 Interpretation of Stress

As stated, a stress value of zero indicates a perfect fit of points from the MDS procedure; however it is not necessary for an MDS based mapping to have zero stress in order to be informative or useful. Stress values close to zero are also tolerable due to an allowable amount of distortion in the model. The cut off value for what range of stress is permissible is largely subjective and may vary depending on the type of data being used or the level of accuracy desired by the researcher. It is also important to note that stress is expected to be higher on data with more samples and more variables, so extreme care must be taken when comparing stress values of data sets with different sizes. There are however, a few broad rules of thumb which may be useful for researchers to remember when performing this method of ordination. The following table summarizes the suggested rules when using STRESS-1.

Table 2.3: Interpretation of STRESS-1

Interpretation	**Stress Value**
Ideal	0 - 0.1
Acceptable	0.1 - 0.15
Unacceptable	0.15 - 1

Of course, even when stress values are sufficiently low to be counted as acceptable, a certain amount of care must be exercised when interpreting the configuration of points. By definition, any amount of stress that is present is due to at least one of the distances between a pair of points being distorted on the MDS configuration (possibly all to some degree). This means that whenever one wishes to make certain conclusions based on specific pairings, it is wise to first check on the accuracy of that specific element of the configuration. Necessary diagnostic tools do exist for this type of checking, and will be elaborated on in Section 2.6. On the other hand, in general, even mappings with higher stress values are potentially able to provide some sense of what small scale patterns exist within the data. That is to say, only patterns when viewing the configuration from a zoomed out perspective might be reliable. Zoomed in observations of intricate details (large scale patterns) would not be reliable. This is due to the fact that longer distances tend to be more accurate than shorter distances on a relative scale. This claim can be verified with the aid of a simple example.

Figure 2.2 demonstrates how longer distances are generally more accurate. In the figure we see the lengths between two different point pairings. These pairings are A-B and D-E with the true distances between points seen to be 3 units and 10 units respectively. The distances between points A-C and D-F however represent the lengths between the original points as seen from an MDS ordination configuration. The MDS calculated lengths for the pairings are 2 and 9 units respectively. In both cases, therefore, the ordination distance is exactly 1 unit short of the true distance. It is simple then to see that in the case of this ordination result, A to C is 66% accurate of A-B while D to F is 90% accurate of D-E. The greater distance therefore exhibits greater accuracy than the shorter when faced with the exact same error.

Using Normalised Stress (NRS) as an example, the effect on the two scenarios is

$$NRS_{AB(C)} = \frac{(3-2)^2}{2^2} = 0.25 \quad \text{and} \quad NRS_{DE(F)} = \frac{(10-9)^2}{9^2} = 0.012$$

Clearly the stress value for the longer distances is much smaller, even

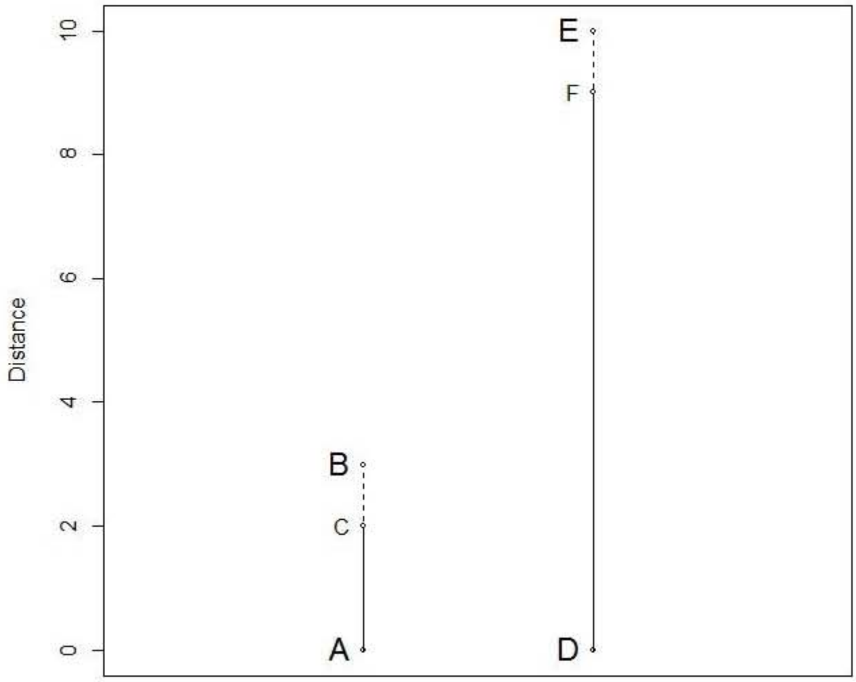


Figure 2.2: Accuracy of Distance

though the deviation is one unit in both cases.

2.5 Dimensionality

The process of Multidimensional Scaling requires a choice of the number of p dimensions in which the configuration of points is to be mapped. As mentioned before, in the event that an MDS procedure results in a final stress value that is unacceptably high, one option available to the researcher is for the procedure to be performed again with a change in output dimension. At this point, it should once again be noted that while in practice the only dimensions that can be visually illustrated and analysed are dimensions one, two and three, MDS is not limited to these. In fact the procedure is able to allocate a configuration of points up to $n - 1$ dimensions; where n is the number of objects/samples in the data set.

When observing stress values for different dimensions usually suggests that as the number of dimensions is increased, the stress value will decrease. Consider a solution $\mathbf{X} : n \times h$ and the dimensions is then increased to $p > h$. All MDS solutions are based on some form of optimisation. If the solution $\mathbf{X} : n \times p$ has a larger stress than $\mathbf{X} : n \times h$, the optimal solution in p dimensions would be $\mathbf{X} : n \times p = [\mathbf{X} : n \times h \quad \mathbf{0} : n \times (p - h)]$ which shows that the stress of the optimal solution will be at least the same as the h-dimensional solution, and possibly smaller. In general, the value of stress will decrease at a decreasing rate as the number of dimensions is increased until p reaches a certain value.

While the above statement is true (increasing dimensions enough is guaranteed to drop stress), realistically stress is likely to be caused, to some extent, by random measurement error. An example illustrating this may be where a researcher is interested in the distances between the tops of buildings that are measured in a hypothetical city center. In this case, it is obvious that the true number of dimensions is three for this set of data (latitude, longitude and altitude) so theoretically a three dimensional MDS configuration should have zero stress. Realistically however, it is most likely that there will be some small amount of stress, which will be due entirely to measurement error. In cases such as these, where the precise number of dimensions is known and applied in the configuration, the value of stress can be used as a direct measure of the accuracy of the data. In this scenario, increasing the number of dimensions will eliminate this stress, in which case the excess dimensions of the MDS configuration will be describing the error of the measurement. Unfortunately, it is very seldom that the case be as clear cut and easily interpretable as this, so such convenient observations are not always possible.

There are two major issues with increasing the number of dimensions in such an ordination procedure. The first problem is obviously that the higher the dimension, the more difficult the interpretation. The most obvious drawback is that any dimension over 3 cannot be graphically portrayed which means that the element of being able to quickly analyse a plot for patterns and groupings be eliminated. The second issue with a large number of dimensions is that with every increment of dimension additional parameters are to be taken into account estimated from the raw data. In the event that say $n-1$ dimensions are used, the complexity of the outcome is for all intents and purposes as complex as the data itself. Having said this, there do exist applications of MDS in which a high number of dimensions is not a problem.

Solutions of Classical Scaling (Principal Coordinate Analysis) are nested, regardless of the p used. That is, $\mathbf{X} : n \times p = [\mathbf{X} : n \times h \quad \mathbf{X}^*]$. This means that the first h coordinate vectors will always be the same no matter what the adjustable p is. Most other MDS methods, which are based on optimisation, are not nested. Therefore, as p changes, the first h dimensions also change.

2.5.1 Euclidean Embedding and Dimensions

As previously described, a number of cases exist which have an effect on the dimensionality of an MDS procedure. The following three cases are defined when the data space is Euclidean and therefore the matrix $\mathbf{D}$ is comprised of Euclidean distances.

The first of these cases is when the distance metric used is itself Euclidean. In this scenario, when $p = m$, the value of stress will be zero. The second case is when the distance calculation is Euclidean Embeddable. This means that despite having definitions different to a Euclidean Distance the distance information is still preserved when depicted in a Euclidean space. In this event, when $p = n - 1$, the value of stress will be zero. The final case is when the distance calculation is Non-Euclidean Embeddable. These cases will never achieve a stress value of zero regardless of the value of p.

2.6 Diagnostic Tools

Multidimensional Scaling is a multivariate tool that, by its very nature, will produce results with an expected amount of distortion or error. It is important for a researcher undertaking tasks by using Multidimensional Scaling to understand the distortion of their output in order to improve on the results. A number of diagnostic tools exist and are used for analyzing the output of Multidimensional Scaling Procedures, some of which will be discussed in this Section.

2.6.1 Scree Plot

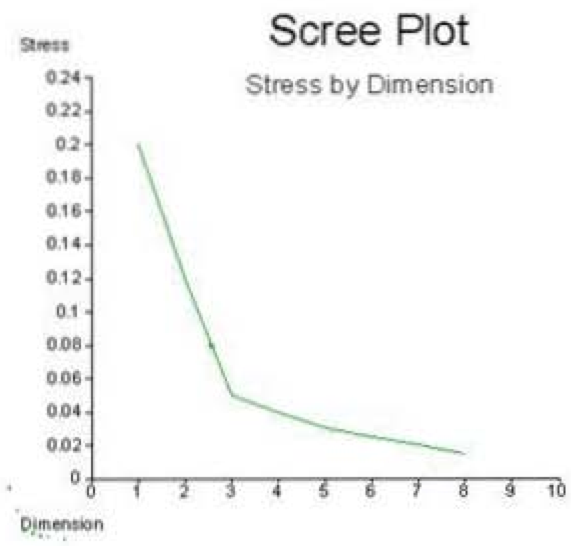


Figure 2.3: Scree Plot Example

The problem of deciding on the number of dimensions to specify for a Multidimensional Scaling procedure on a given set of data can be made easier with the use of a *Scree Plot*. A Scree plot is achieved by running MDS on the data a number of times, *ceteras paribus*, with only the number of output dimensions changing. The resulting stress value is recorded after each run. The Scree Plot thus portrays the curve of plotting stress versus dimension.

As stress has already been described to decrease as dimensions increases, the plot demonstrates a decreasing monotonic function. The appropriateness of the dimensionality of the data is revealed by the shape of the rate of decline of stress. Figure 2.3 shows an arbitrary example of a Scree Plot with an easily interpretable shape. The Scree Plot example shows an obvious kink in the curve, otherwise referred to as an "elbow" in the plot. It is at this point that an indication of the true dimensionality of the data is revealed.

This elbow is obviously the point where increasing the dimensions yields a diminishing return on the stress, and thus the most appropriate MDS model has as many dimensions as the number of dimensions at the elbow. Returning to the example of measuring the distances between building tops. If this study were to be performed, it is quite likely a resulting Scree Plot would look very much like the example shown in Figure 2.3. The example shows the value of stress dropping steeply from the first dimension until the third, and subsequently the curve flattens out. The 'elbow' of the curve very clearly occurs when p is three and thus, as expected, $p = 3$ is the most appropriate number of dimensions to use in the MDS procedure. As mentioned in the previous Section, it is highly likely that some element of measurement error occurs which are explained by all dimensions greater than 3. The stress values for all $p > 3$ in the Scree Plot allow for visual inspection of the nature of this stress due to measurement error.

Figure 2.4: Scree Slope

The derivation of the naming of the Scree Plot may be of interest to some readers. The word 'scree' is a geological term which refers to the build up of debris which collects at the lower part of a rocky slope. The resulting slope is called the 'Scree Slope'. The image shown in Figure 2.4 gives an example of such a scree slope and provides some insight into why the Scree Plot has been aptly named.

The portion of the curve that occurs to the right hand side of the elbow is effectively the scree slope, and the 'debris' is made up of what has been called factorial scree. This refers to the non systematic noise and measurement error that the further dimensions account for.

In reality, such clear elbow points in the scree plot are not always so obvious. This usually means that a certain amount of subjective interpretation is often required in interpreting such Scree Plots.

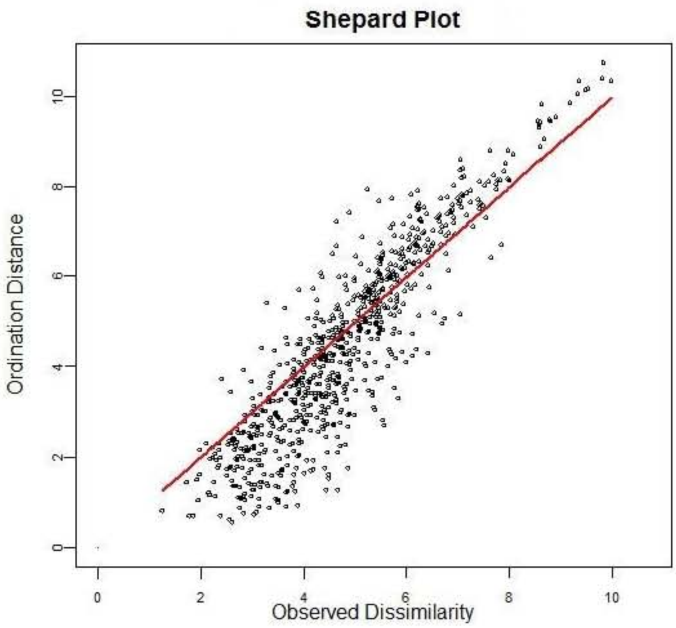
2.6.2 Shepard Plot

Another tool for judging the accuracy of an MDS procedure is the Shepard Diagram (Shepard, 1962). The diagram is a two dimensional scatter plot with the X-axis corresponding to the input proximities (δ_{ij}) and the Y-axis corresponding to both the MDS distances (d_{ij}) and the transformed proximities ($\hat{d}_{ij}$). Each point on the diagram represents a pairing of the subjects in the data. This means that the Shepard diagram will have $\binom{n}{2}$ points, where n is the number of objects in the data. The Shepard diagram is therefore laid out in such a way that an observer is able to assess the accuracy of the ordination configuration with regards to every pairing individually. Since the observed distance will never change, each point will be isolated to a single vertical line, and therefore, the point will move up and down the line depending on the result of the ordination. Examples of the Shepard Plot are found in Figures 2.5(a) and 2.5(b). In both instances, the information is based on an MDS result where the data has 40 objects, and therefore 780 points.

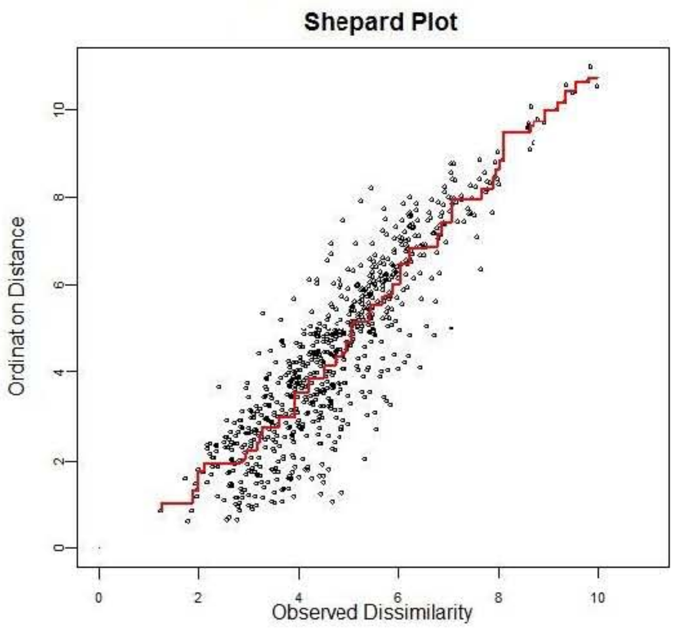
The error of each point in the Shepard Diagram can be directly assessed by measuring the vertical distance between the actual location of the point and its hypothetically correct location. This hypothetically correct location depends on the type of MDS being performed. In metric MDS, where observed distances require matching, the distance between a pairing has been represented exactly when its X and Y coordinates are equal as the observed distance equals the ordination distance. Continuing this logic, a configuration that has a perfect fit will have all the points in a straight line along $y = x$. Figure 2.5(a) is an example Shepard Plot from a metric MDS result. Included in the plot is a line through the diagonal, indicating the optimum fit. A configuration with a poor fit will have a Shepard Plot with points deviating far from this line. In Non-Metric MDS however, the relaxed metric assumption means that ordination distances do not strictly need to match the observed distances. Instead only the ordering of points is taken into account. In these non-metric instances the representation is exact when the observed distance d matches the transformed disparity $\hat{d}$. This means that on the Shepard Diagram it is the vertical distance been the point and the transformation curve that shows the amount of deviation in the representation of the object pairing. Figure 2.5(b) shows a Non-Metric MDS based Shepard Diagram. The transformation curve in this case is a step function derived from an isotonic regression transformation.

The most useful element of the Shepard Diagram as a diagnostic tool is that, unlike evaluating a stress value, it is clear to see where the deviations in the predicted configurations lie. This allows the researcher to easily pick up on clear outliers and identify possible systematic deviations. It is also useful to note that the diagram shows the composition of the stress value. For metric MDS the sum of squared vertical distances between the points and the diagonal line forms the common component of most stress formulas. In non-metric cases, it is the sum of squared vertical distances between the points and their fitted values, represented by the transformation curve.

Figure 2.6 gives the example of how the Shepard Diagram can be used for such diagnostic purposes. Both plots show information on the exact same data as depicted in Figure 2.5(b) except in each scenario a single point has been positioned very inaccurately. Inspection of the Plot in 2.6(a) reveals exactly 39 points are positioned very obscurely and can certainly be seen to be well out of place. A researcher would confirm quickly that these points in fact relate to each of the 39 pairs that the single misplaced point has with the other objects and identify it to be a point of concern. The researcher would also be able to determine the nature of the inaccuracy. Since it is clear that for each of the 39 pairings of interest, the ordination distance is longer than the observed distance. This would imply the point is positioned outside the group and not incorrectly within it. If alternatively the point were to be severely misplaced within the group, a situation like that shown in Figure 2.6(b) would arise. This Shepard Diagram reveals a clear misplacement of points above and below the majority of points. The

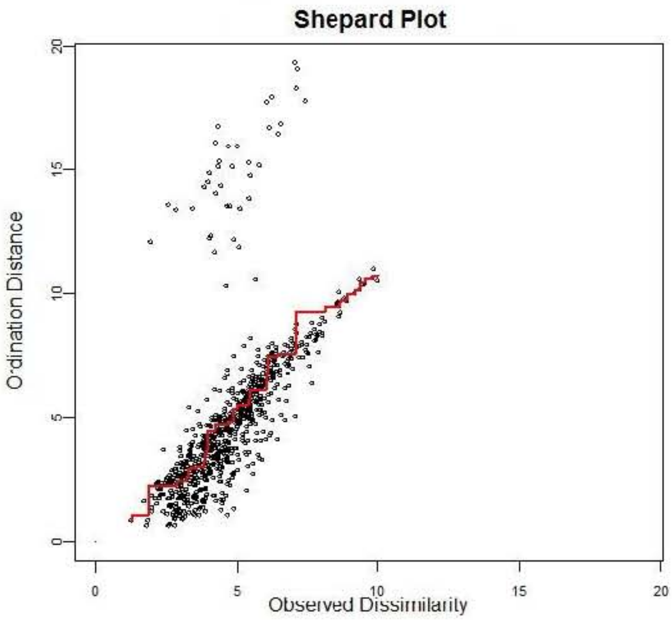


(a) Metric Shepard Diagram

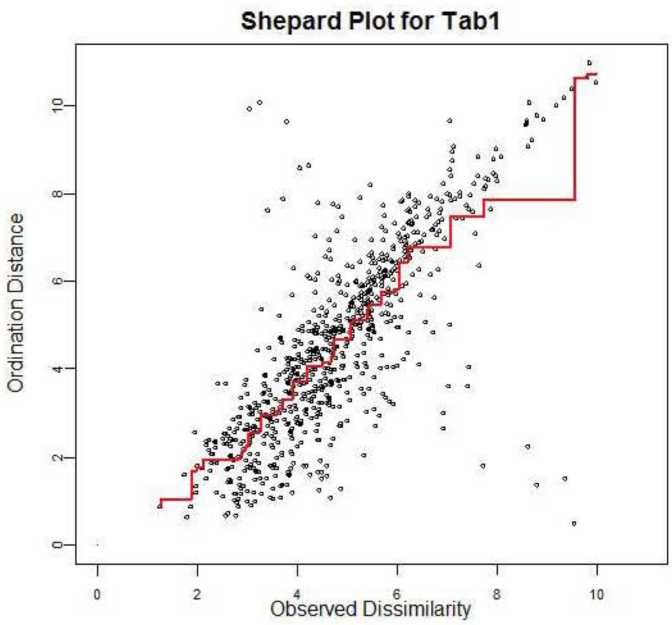


(b) Non-Metric Shepard Diagram

Figure 2.5: Shepard Diagram Examples



(a) Out of Group Distortion



(b) In Group Distortion

Figure 2.6: Example of Shepard Plots With One Distorted Point

points found below the others would show which pairings have been placed closer together than what would have been ideal. See also Section 5.4.3 for further illustration of interpretation of the Shepard Plot.

The visual look of the Shepard diagram is different depending on the type of data being used as well as the format of Multidimensional Scaling. The major variation in look of the diagram depends on the type of proximity matrix used. When the proximity matrix has information regarding the similarities between the items, the Shepard Plot will have a negative slope of points. The logic behind this is intuitive, since the X-Axis will represent the similarities of the data, and the Y-Axis remains representing distance between points in the ordination configuration. This implies that pairs that have a high similarity have a small ordination distance and pairs with low similarity will have longer ordination distances. Thus a negative slope. On the other hand, when proximities are dissimilarities the plot will have a positive slope. The logic to this is similar. The plot itself will also give an idea, to the trained eye, of the type of MDS that has been used. In metric scaling the line of points will be straight. In non metric scaling, however, the points will form a weakly monotonic function.

2.7 Data

Multidimensional Scaling is applicable to data in any of a number of forms. While some data sets are comprised of n objects and m variables, other factors do come into effect that may influence how MDS should be approached. For example, the formal definition of number of 'modes' and 'ways' of the data may alter the most appropriate form of MDS to be utilised. These and various data classifications and transformations will be discussed in this Section.

2.7.1 Types of Data

Several classifications of data exist that vary depending on how data is recorded and what it represents. Four major scale classifications of data have been identified and are listed below.

2.7.1.1 Nominal Scale

Data collected using the nominal scale represents only categorical information. If numbers are chosen to represent certain categories they will have no numerical interpretation. Letters are just as appropriate to represent the different classes.

2.7.1.2 Ordinal Scale

Ordinal data is data that can be ranked but does not hold any quantitative property. Recording the positional results of a race is an example of ordinal data. It can be observed that 1st place was faster than second, but no information regarding the extent of difference is given.

2.7.1.3 Interval Scale

Data collected on the interval scale is quantitative by nature and thus, unlike the previous two scales, exhibits continuous qualities. Therefore under the Interval scale, the difference between points is meaningful and the extent of the difference is recorded.

2.7.1.4 Ratio Scale

The final scale is the Ratio Scale, which has a continuous nature like the Interval Scale, but also is defined by a relevant zero point. Variables measured on the Ratio Scale can therefore never be less than zero. For example temperature measure in degrees Celsius is not measured on the ratio scale since negative temperatures exist. Temperature measured in Kelvin however is on the ratio scale since zero Kelvin represents the lowest possible temperature.

2.7.2 Modes and Ways

The four scales determine how the data is measured, but do not necessarily describe the data as a whole. Two definitions that do make these general descriptions are termed 'modes' and 'ways'. Modes are defined as the number of sets of objects that occur within one data set. Cox and Cox (2001) uses the example of data being gathered from a number a judges tasting a number of whiskies and comparing them to one another. In this scenario, the data would be two mode, with the judges and the whiskies being the two modes. The ways of a data set is defined by the number of indexes that exist between objects. In this whisky judging example, there are three ways, as the three index's are the two whiskies being compared and the judge who compared them.

The most common form of proximity data used in Multidimensional Scaling is one mode and two way, i.e. the whisky example with only one judge.

2.7.3 Transformations of Data

Non-Metric Multidimensional Scaling is appropriate when only the rank ordering of the dissimilarities is relevant to the data. In these cases it is desirable to assign numerical values to the proximities in such a way that these values, the $\hat{d}$ disparities, exhibit the same rank order as the data (Groenen and van de Velden, 2004). These methods of assigning appropriate $\hat{d}$ values are the transformations of data relevant to MDS. Therefore, in Non-Metric MDS, the process is simultaneously required to obtain a configuration $\mathbf{X}$ and the corresponding $\hat{\mathbf{D}}$ matrix with each iteration. The transformation that will be used exclusively throughout this dissertation will be the isotonic regression transformation, sometimes referred to as monotone regression or ordinal transformation. This process will be described and illustrated in Section 3.3.

Other examples of transformations of the data to the $\hat{d}$ disparities are (Borg and Groenen, 2005): the ratio transformation, the interval transformation, logarithmic transformation, exponential transformation and spline transformation.

2.8 Measures of Proximity

The proximity matrix has been mentioned numerous times thus far and this is because it plays a vital role in Multidimensional Scaling. The proximity matrix, whether it be in similarity or dissimilarity format, forms the primary input of any MDS procedure. This Section will discuss the derivation of both similarity and dissimilarity's within the context of Multidimensional Scaling.

2.8.1 Measuring Dissimilarities

The use of dissimilarities as the proximities in MDS is probably the most common. For the sake of generality, dissimilarities as proximities will be the primary form of proximity referred to in this dissertation and will be used exclusively in the MDS-GUI to be discussed in Chapters 5 and 6. Dissimilarities between objects of interest can be measured directly and therefore a dissimilarity matrix constructed manually. It is, however, quite possible and more common to calculate the distances between objects in an $n \times m$ data set with one of a number of different dissimilarity measures. A selection of these measures of dissimilarities are presented here, and will be referred to regularly throughout the remainder of this dissertation.

2.8.1.1 Euclidean Distance

The Euclidean Distance measurement is the measure that most people associate with the word "distance". The derivation of the formula is based on the Pythagorean principals.

$$\delta_{rs} = \sqrt{\sum_{i=1}^{m}(x_{ri} - x_{si})^2}$$

The Euclidean Distance is, as stated, the most common form of measurement used when calculating distance proximities from data for MDS methods. In addition, it is the measurement most often used for evaluating the resulting MDS configurations in terms of calculating the **D** matrix used in the various formulas of stress. This, as with so many of the calibrations of MDS, is the norm but not the rule. The Euclidean Distance as a measurement is simply used most often due to its mental visualization and ease of use to most researchers.

2.8.1.2 Weighted Euclidean Distance

The Weighted Euclidean distance measurement is, as the name suggests, an extension of the Euclidean Distance. The Weighted Euclidean measurement formula incorporates an additional factor in its calculation, being a weight component where each of the m variables in the data have an individual weighting.

$$\delta_{rs} = \sqrt{\sum_{i=1}^{m} w_i (x_{ri} - x_{si})^2}$$

This feature may be useful in a number of different ways. Firstly, it allows the researcher to place more or less importance on certain variables if it is appropriate to the situation. Alternatively in the event that there are variables with very high variance, a lower weighting may be applied and *vice versa* for when variance is very low.

2.8.1.3 Mahalanobis Distance

The Mahalanobis Distance was developed by an Indian statistician, Mahalanobis (1936). The measurement makes use of the covariance matrix and therefore has uses when there are suspected patterns based on the correlations between variables in the data.

$$\delta_{rs} = \sqrt{(\mathbf{x}_{r:(n\times1)} - \mathbf{x}_{s:(n\times1)})^T \boldsymbol{\Sigma}^{-1}_{(n\times n)} (\mathbf{x}_{r:(n\times1)} - \mathbf{x}_{s:(n\times1)})}$$

This measurement of distance has a wide range of applications, but has been found to be particularly useful in cluster analysis and the detection of outliers (Mahalanobis, 1936).

2.8.1.4 City Block Metric

This distance measure was initially conceptualized by Minkowski in the early 1900's, (Evans, 2010) and has alternative names, such as the Manhattan Distance and, more recently the L1 distance (Cha, 2007). It will however be referred to as the City Block distance in this dissertation and has the form shown below. The City Block Metric assigns importance to only mutually orthogonal distances, unlike the Euclidean Distance which assigns equal importance to all directions (Heiser, 1988).

$$\delta_{rs} = \sum_{i=1}^{m} |x_{ri} - x_{si}|$$

A property of this metric is that a set of points with equal distances to a central point creates a square; whereas the corresponding Euclidean situation reveals a circle.

2.8.1.5 Minkowski Metric

Another of Minkowski's metric distance conceptualizations was named after himself, and is called the Minkowski metric. This metric has generalised properties of both the Euclidean Distance and Minkowski's own City Block Metric.

$$\delta_{rs} = \sqrt[\lambda]{\sum_{i=1}^{m} w_i |x_{ri} - x_{si}|^{\lambda}}$$

This measure of distance has had particularly noteworthy use in the area of equations of motion in the field of Physics (Evans, 2010).

Further observation reveals that three special cases of the Minkowski Metric exist. These three special cases are as follows:

1. In the event that $\lambda = 2$ and $w_i = 1$, the Euclidean Distance formula is achieved.

2. In the event that $\lambda = 2$ and $w_i \neq 1$, the Weighted Euclidean Distance is achieved

3. In the event that $\lambda = 1$ and $w_i = 1$, the City Block Metric is achieved.

2.8.1.6 Canberra Metric

The Canberra Metric was first conceptualised by Lance and Williams (1966). The measurement of distance is a weighted version of the City Block Metric.

$$\delta_{rs} = \sum_{i=1}^{m} \frac{|x_{ri} - x_{si}|}{(x_{ri} + x_{si})}$$

The metric is defined only when x_{ri} and x_{si} are non-negative, and most importantly when $x_{ri} + x_{si} \neq 0$. The measure of distance has particular use in the measure of disarray for ranked lists, with most notable applications in functional genomics (Jurman et al., 2009).

2.8.1.7 Bray-Curtis Distance

This measure of distance, developed by Bray and Curtis (1957), is specifically a non-metric measurement. This means that the dissimilarity measurement does not preserve the ratios and intervals of the distances, only the order.

$$\delta_{rs} = \frac{1}{p} \frac{\sum_{i=1}^{m} |x_{ri} - x_{si}|}{\sum_{i=1}^{m} (x_{ri} + x_{si})}$$

The Bray-Curtis Distance is known for its robust and reliable results with applications mostly in ecology and other natural sciences (Schulz, 2007).

2.8.1.8 Soergel Distance

The Soergel metric is strictly for non-negative data.

$$\delta_{rs} = \frac{\sum_{i=1}^{m} |x_{ri} - x_{si}|}{\sum_{i=1}^{m} max(x_{ri}, x_{si})}$$

The metric has had its most notable application in discovering efficiency hammock paths (Hossain et al., 2010).

2.8.1.9 Bhattacharyya Distance

A. Bhattacharyya was a colleague of Mahalanobis and developed this metric measurement in 1943 (Bhattacharyya, 1943).

$$\delta_{rs} = \sqrt{\sum_{i=1}^{m}(x_{ri}^{\frac{1}{2}} - x_{si}^{\frac{1}{2}})}$$

The measure has use for comparing samples in a wide range of fields of application, including computer science and biology.

2.8.1.10 Wave-Hedges Distance

The Wave-Hedges metric assigns higher weights to large relative distances.

$$\delta_{rs} = \frac{1}{p}\sum_{i=1}^{m}\left(1 - \frac{min(x_{ri}, x_{si})}{max(x_{ri}, x_{si})}\right)$$

The metric has been found to have uses in content based image retrieval systems (Kamiechetty et al., 2002).

2.8.1.11 Angular Separation

Angular Separation has a slightly different take on distance measurement. The form of measure usually observes the distance between two points measured from a single reference point.

$$\delta_{rs} = 1 - \frac{\sum_{i=1}^{m} x_{ri}x_{si}}{\sqrt{[\sum_{i=1}^{m} x_{ri}^2 \sum_{i=1}^{m} x_{si}^2]}}$$

Angular Separation is most commonly used in the measures of the distance between celestial bodies from the Earth. For astronomical applications, the result is usually converted to degrees or radians.

2.8.1.12 Divergence

The Divergence formula is given by:

$$\delta_{rs} = \frac{1}{p} \sum_{i=1}^{m} \frac{(x_{ri} - x_{si})^2}{(x_{ri} + x_{si})^2}$$

where $x_{ri} + x_{si} \neq 0$. Applications for the Divergence distance have been found in using it as a discrimination measure for hidden Markov models (Silva and Narayanan, 2006).

2.8.2 Measuring Similarities

Proximities in the form of similarity matrices are also possible within the scope of Multidimensional Scaling. It is more common to accumulate the proximity data manually when measuring the similarities between objects than when gathering information for dissimilarities. This is usually done by, preferably objectively, assigning each pairing a score from a range where, say, a high score indicates very similar and a low score indicates not at all similar. Once again, the resulting proximity matrix is often symmetric, but this need not be the case.

It is often desirable for a researcher to transform the similarity matrix into a dissimilarity matrix for ease of the MDS computations. As mentioned before, in the MDS-GUI transformation will always be desirable and is automatically performed (to be discussed later). A number of ways exist that allow for this transformation. In the event that the scoring of the similarities is done on a range between 0 and 1, a direct conversion can be made. For example the following three conversions are possible.

$$\delta_{rs} = 1 - s_{rs}$$

$$\delta_{rs} = \sqrt{1 - s_{rs}}$$
$$\delta_{rs} = \sqrt{1 - s_{rs}^2}$$

where s_{rs} is the similarity score between the r^{th} and s^{th} objects. This of course preserves the 0-1 range for the dissimilarity scores. If on the other hand the similarity scoring is not done on a range between 0 and 1, there are further options. The researcher is able to scale the proximities to between 0 and 1 and proceed as before, or else they are able to use the maximum score as the upper bound, thus:

$$\delta_{rs} = max(s) - s_{rs}$$

This transformation only holds when $\forall s_{rs} \geq 0$ as should be expected when gathering similarity data.

Similarity measures are also often appropriate when dealing with binary data. Similarity coefficients from binary data are calculated with any of a number of different methods, two of which will be discussed in this Section. Before these coefficient formulas are addressed, a brief explanation of the terms in the formulas must be provided. This is done using the following simple table, which has been formulated by (Cox and Cox, 2001).

Table 2.4: Binary Data Similarity Coefficient Key

Object r	**Object s**		
	1	**0**	
1	a	b	$a + b$
0	c	d	$c + d$
	$a + c$	$b + d$	$m = a + b + c + d$

Table 2.4 shows how the components a, b, c and d are made up for any two objects, say r and s, when the total number of binary variables in the data is equal to m. They are defined as follows: a is defined as the number of variables that have 1 for both s and r; b is the number of variables that are 1 for s and 0 for r; c is the number of variables that have 0 for s and

1 for r; and finally d is the number of variables that have a score of 0 for both s and r. With these definitions established, the following similarity coefficients may now be discussed.

2.8.2.1 Jaccard Coefficient

The Jaccard Coefficient is defined as the intersection between two sets divided by the union of the sets. In terms of binary data with the descriptions demonstrated by Table 2.4, the Jaccard Coefficient is given by the following formula:

$$s_{rs} = \frac{a}{a+b+c}$$

The coefficient is widely used in measuring the similarity of data sets and does so with the proportion of positive matches over total instances. The Jaccard Coefficient is thus applicable to cases where only positive matches are relevant.

2.8.2.2 Simple Matching Coefficient

The Simple Matching Coefficient is an extension of the Jaccard Coefficient and differs by the fact that while the Jaccard Coefficient only accounts for shared positives, the Simple Matching Coefficient accounts for both positive and negative matches. The formula is given below:

$$s_{rs} = \frac{a+d}{a+b+c+d}$$

The formula demonstrates the Simple Matching Coefficient is comprised of the ratio of total matches (positive and negative) over total instances. The Simple Matching Coefficient is applicable when any form of match is significant.

2.8.2.3 Correlation

A form of measurement that has proven to be popular when using MDS is that of the correlation between objects. This popularity no doubt stems from the fact that the concept of correlation is known widely throughout all scientific fields and is very easy to calculate for large sets of data, thanks to modern software. Correlation is another statistic that is often referred to generally, however a number of forms of correlation do exist: three of which will be discussed. These three correlation methods are: Pearson's Correlation, Spearman's Rank Correlation and Kendall's tau. The Pearson's Correlation is defined as the covariance of two variables divided by the product of their standard deviations.

$$Pearson's : s_{rs} = \frac{\sum\limits_{i=1}^{m}(x_{ri} - \bar{x}_r)(x_{si} - \bar{x}_s)}{\sqrt{\sum\limits_{i=1}^{m}(x_{ri} - \bar{x}_r)^2 \sum\limits_{i=1}^{m}(x_{si} - \bar{x}_s)^2}}$$

The Spearman's Rank Correlation is an extension of Pearson's Correlation and has been adapted to specifically deal with ranked data. The format of the formula is unchanged from the Pearson's Correlation, with the only difference being that all raw data is converted to rank scores. The following equation is exactly the same as before, with the exception that $R(x_{ri})$ is defined as the rank of the i^{th} element of the r^{th} variable and $\overline{R(x_r)}$ is defined as the mean rank of the r^{th} variable.

$$SpearmansRank : s_{rs} = \frac{\sum\limits_{i=1}^{m}(R(x_{ri}) - \overline{R(x_r)})(R(x_{si}) - \overline{R(x_s)})}{\sqrt{\sum\limits_{i=1}^{m}(R(x_{ri}) - \overline{R(x_r)})^2 \sum\limits_{i=1}^{m}(R(x_{si}) - \overline{R(x_s)})^2}}$$

The Kendall's Tau Coefficient is used in testing for independence between variables and is applicable to ranked data. The formula below requires some explanation. The term 'concordant pairs' refers to the number of corresponding joint pairings between two ranked vectors that are in agreement. For example (Bolboaca and Jantschi, 2006) if the pairs, (x_{ri}, x_{rj}) and (x_{si}, x_{sj}), are compared between variable r and variable s, they are said to be concordant if either $R(x_{ri}) > R(x_{rj})$ and $R(x_{si}) > R(x_{sj})$ or $R(x_{ri}) < R(x_{rj})$ and $R(x_{si}) < R(x_{sj})$. Conversely, the pairs are termed 'discordant' if neither of these hold. The total number of concordant pairs is given by C and the total number of discordant pairs is given by D. The resulting Kendall's Tau Correlation formula is given by:

$$Kendall'sTau : s_{rs} = \frac{C - D}{\frac{1}{2}n(n-1)}$$

The values of the resulting proximity matrices from all three of these formulas will be between 0 and 1, with higher values indicating greater similarity between objects. It can therefore be treated exactly the same as any other similarity matrix with that range of values.

2.9 Setting up of MDS Procedure

Section 2.3, entitled *The General MDS Algorithm*, made reference to a number of adjustable parameters that are involved in most Multidimensional Scaling procedures. While most of these parameters have been elaborated on in previous sections, this Section will summarise the decisions a researcher will need to make when performing MDS procedures. This will be done with particular emphasis on the choices that must be made before a first attempt at MDS. Each of these parameters is expected to affect the final coordinate configuration in some way, albeit some tend to have a greater influence than others.

Firstly, before any form of ordination can be attempted, it is naturally important that the multidimensional data be decided upon. Referring to Section 2.8, it can be seen that there are a number of choices that must be made with regards to the proximity matrix used in the MDS process. These choices are namely: deciding whether to form the proximity matrix directly (manual counting) or indirectly (proximity calculations); having the proximities as similarities or dissimilarities; and finally accepting a symmetric form of the matrix or allowing asymmetry amongst the proximities. In the event that an asymmetric proximity matrix is accepted, it should be noted that this will double the number of points in the configuration and increase the number of points in the Shepard Diagram to $\binom{2n}{2}$. For convenience, it will be assumed that all proximity matrices are symmetric throughout the remainder of the dissertation.

With the input data decided upon, the crucial decision of which form of Multidimensional Scaling must be made. In many cases, the form of the proximities will have an influence on the type of MDS to be used. If the proximities place importance on the extent of the dissimilarities, then a metric MDS is appropriate, while a non metric MDS should be used if only rank order of the proximities are relevant. Within each of the categories there are further choices of the specific MDS, all of which will be covered in Chapter 3. The researcher may want to assess the results of a number of the various methods for comparison.

The output of the ordination then needs to be defined in terms of p, the number of dimensions in which the points will be mapped. The concept of the number of dimensions on which to derive the MDS configuration is discussed in Section 2.5 and the Scree Plot in Section 2.6. The researcher may have a preexisting idea of the most appropriate number of dimensions for the specific data, and if not, the Scree plot will be useful in determining what p should be. In practice, a common method is to first perform MDS in two dimensions and then make an assessment of whether additional dimensions will be appropriate or necessary. Some care must be put into the choice of p as the ordination is sensitive to it. If too few dimensions are used, multiple axes of variation will be portrayed by a single ordination dimension, whereas too many dimensions will cause a single axis of variation to be portrayed over many dimensions of ordination (Holland, 2008). In order to explain this, the example of mapping the distance between the tops of buildings is revisited. Three axes of variation for the scenario could be seen as the axis of latitude, the axis of longitude and the axis of altitude. In the event that a one dimensional ordination is performed, the three axes of variation will be portrayed on a single dimension. If alternatively, the ordination is done in more than three dimensions, the three axes of variation are portrayed by too many dimensions. In both cases the nature of the differences between the building tops is nearly impossible to determine by observing the MDS configuration. This example makes very literal use of the word 'axes': realistically the axes of variation may require somewhat more thought in their interpretation, such as aggressiveness when exploring data on breeds of dog. Details on how to interpret these axes of variation will be provided in Section 2.10. A rule of thumb suggested by some specialists of MDS is that a p dimensional ordination requires at least *4p* variables in the raw data (Wickelmaier, 2003). This is clearly just a guide line however, as a data set consisting of only six or seven variables should still be able to have $p = 2$ without any problem. This is despite the rule specifying that there should be at least eight variables for this ordination output.

The starting configuration should then be provided. With the exception of Classical Scaling, every form of MDS requires an initial configuration of n points in the p dimensional space. The starting configuration is a parameter that tends to have a substantial influence on the final configuration of the result. The default starting configuration is often accepted to be the result of a Classical Scaling on the data, however it may be of interest to instead use a completely random set of coordinates or perhaps the resulting configuration obtained by some other method of ordination. The starting configuration is usually something that should be experimented with, as it should be noted how susceptible a set of data is to different starting positions.

The final parameters that need to be set are those that correspond to how the progress of the MDS process is being monitored. The form of stress analysing the configuration of each iteration must be defined and the tolerance of differences between stress of iterations needs to be set. Section 2.4 provides a few examples of the variation of stress formulas, namely: STRESS-1, STRESS-2 and Normalised Raw Stress. With regards to the tolerance: the higher the value the less the number of iterations and therefore, the possibility of a less reliable result. Alternatively a very small level of tolerance will imply more iterations and the probability of a more reliable configuration, however the procedure becomes more time consuming and more computationally intensive. The most appealing trade-off between the two extremes is a tolerance that is small enough to bring about an adequately converged result where if the value were to be lowered the change in the result would be negligible. In practice a tolerance of say 0.001 tends to be more than adequate.

2.10 Interpretation of MDS Results

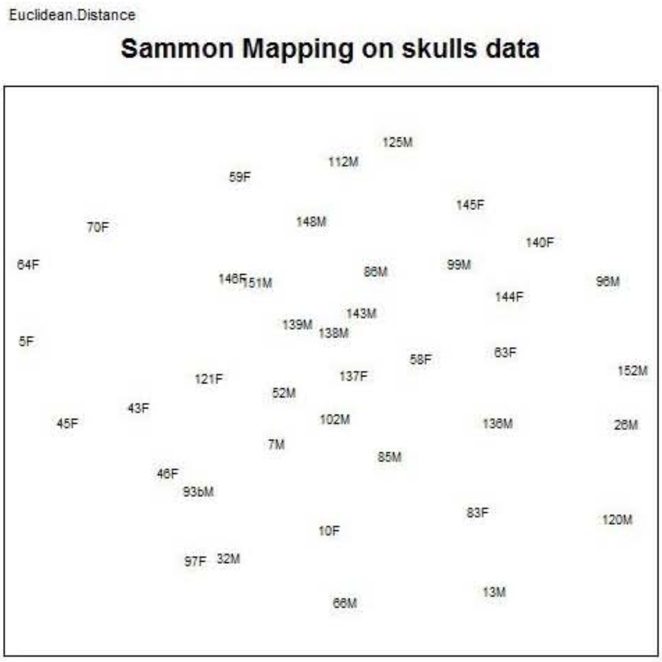
An important skill required by any researcher attempting to use Multidimensional Scaling of any form is the ability to interpret the results of the ordination process. This does not only entail making judgments on the relationships of subjects based on their position in the final mapping, but also assessing these observations in conjunction with all other outputs of the procedure. The diagnostic tools discussed in Section 2.6 play an important role in interpreting MDS results, as do the axes of variation mentioned in Section 2.9. This Section will explore the various ways of interpreting the output of the ordination procedure. It should be noted that while various methods are discussed, they should all be used to compliment one another and no one aspect should be assessed in isolation. It should also be noted that many of the analytical techniques referred to in this Section require some form of computer software with MDS specific applications. The MDS-GUI R-package, of which Chapters 5 and 6 are the subject, is one such software that has this functionality. A list of other MDS related software is also provided in Section 5.7. The Skulls data set (Fawcett, 1901), described in the Appendix, is used primarily in this Section for the purpose of demonstration of interpretation.

2.10.1 Interpreting Configuration

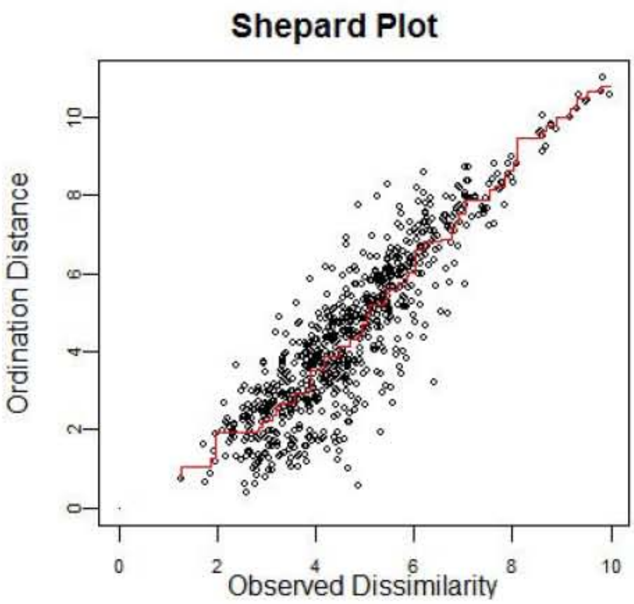
An MDS procedure that is performed where p is either one, two or three will produce a configuration that may be graphically portrayed, albeit that a three dimensional configuration is likely to need slightly more advanced graphical software. These visualised configurations will entail n points, where n is the number of objects in the data set. Two other outputs from an MDS procedure that are vital in interpretation are the value of stress and the Shepard Diagram. The resulting stress value should be scrutinised before any real analysis of the configuration is to take place. Only results with stress values low enough need even be considered for further analysis. The reader is referred to Section 2.4.6 for guidelines on stress interpretation. The Shepard Diagram however, should be assessed in conjunction with the configuration, as any observation made on a pairing illustrated by the configuration must be cross checked with the corresponding point on the Shepard Diagram. Therefore, a major claim based on the supposed relationship between two points is likely to be invalid if the Shepard Diagram reveals it has been inaccurately portrayed.

Figure 2.7 shows a simple non-metric MDS result, using Sammon Mapping, on the *Skulls* data. The corresponding STRESS-1 value for this particular result is roughly 0.18, which is considered slightly unsatisfactory. Under normal circumstances the high stress value would warrant further steps be taken, in terms of adjusting parameters of the procedure, however since this has purely demonstrative intent, the current solution will be retained. The figure shows the resulting configuration of the procedure (Figure 2.7(a)) and the corresponding Shepard Diagram (Figure 2.7(b)). Inspection of the configuration reveals only basic observations of the relative relationship between individual points, where points close together are seen to be similar and those further apart are dissimilar. The Shepard Diagram also displays a fairly generic output with the majority of the points lying close to the diagonal, showing accurate fit. The diagram does however also show a number of points lying above the majority trend and other lying below it. This suggests that a number of points have been either under or over stated compared to their true distance. However, considering the configuration resulted in a high stress value, this observation is unsurprising.

When analysing the configuration there are two major concepts that must be remembered, both of which have been discussed already but deserve reiteration. The first of these is that the orientation of the points is completely arbitrary: and following this, the axes of the plotting area are meaningless. The researcher is thus able to rotate, reflect and zoom in and out of the configuration without losing any information, provided that the distances between the points remain constant on a relative scale. The second concept is that most distances between points are usually distorted with consideration that longer distances are less distorted. One general explanation



(a) MDS Configuration



(b) Shepard Plot

Figure 2.7: Skull Data Example

is provided for this in Section 2.4.6, where it was shown that the Normalised Raw Stress of a longer distance, with an inaccuracy of 1 unit, has a smaller stress value than a shorter distance with the same inaccuracy. In the specific example shown the stress value of the scenario with longer lengths was almost half that of the scenario with shorter lengths. Another valid reason exists, however, when MDS is performed by an MDS computer program. Since all *stress* calculations are based on a sum of squared difference component, it follows that bigger discrepancies have the higher influence on *stress*. Since longer distances have the greater potential to be inaccurate by greater amounts, most MDS algorithms and softwares account for this and focus on achieving higher accuracy on longer distances. Shorter distances can be inaccurate by the same proportion and have much less influence on the *stress* value. They therefore require less attention by the algorithms.

Ease of interpretation of a configuration may often rely on how the researcher has set up the procedure and categorised their data. In the event that the objects in the data set fall under specific categories, it is suggested that each category be assigned a different colour. For example, if the subjects of the data set are people, one might consider having the points representing males being one colour and those representing females another. Alternatively, if gender is a non important factor in the study, age groups could be considered important and each group given a different colour, etc. The benefit of colour coding categories is that any notable differences between categories should be picked up from the MDS configuration straight away by the researcher. More subtle differences may take slightly longer to identify and interpret, however the presence of colours will make this visual component more manageable. A more in-depth discussion on the different forms of colour coding can be found in Section 4.2.2.7, of which the 'RColorBrewer' package is the subject. Figure 2.8 demonstrates the same configuration as before but with the inclusion of category differentiation. The skulls identified to be from males have been coded blue while those from females have been coded pink. Immediately it is noticed that the different genders tend to different sides of the configuration; an observation that may not be as clear without the addition of colours.

Another feature to be observant of is that of clusters of points within

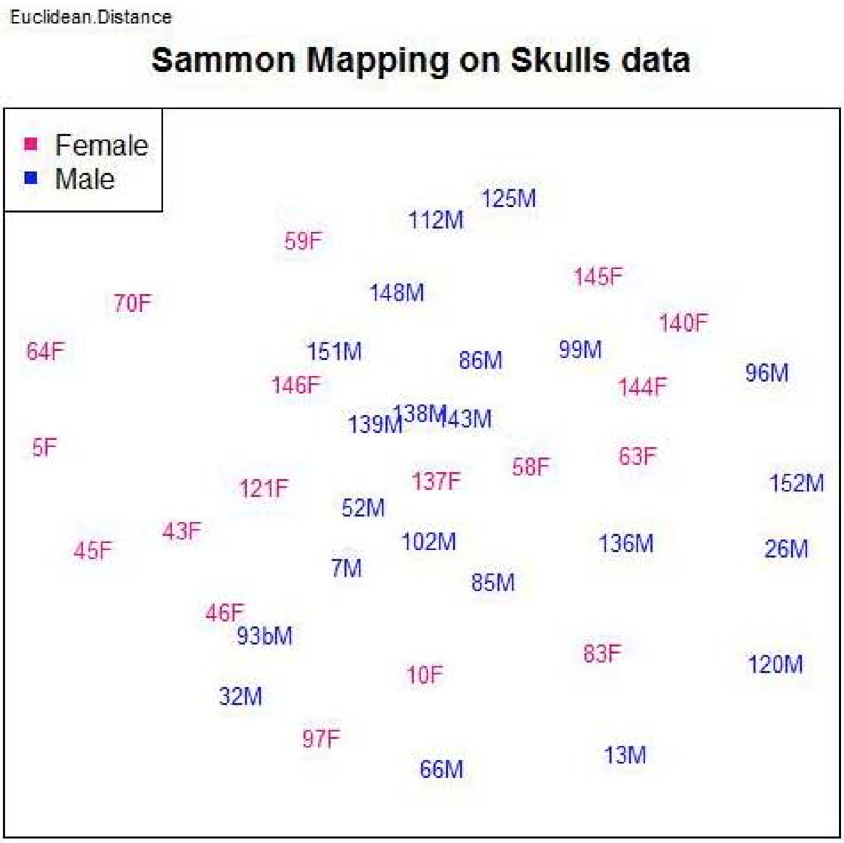


Figure 2.8: Skull Data With Coloured Categories

the data. Clusters can be highly informative in a sense of suggesting which groups of subjects have exhibited strong relationships to one another. Sometimes certain clusters are to be expected and thus serve useful in confirming the similarities, and in other cases the presence of a cluster may provide new information and suggest relationships that were not previously suspected. One vital concept that researchers must be vigilant of when interpreting clusters is that only the presence of clusters may be commented on and not the internal relationships of points within clusters. This follows from the fact that shorter distances tend to be more inaccurate and tight clusters are made up primarily of short distances. Internal analysis of clusters should be carried out by extracting sub matrices from the data which will allow further MDS procedures to be performed on the cluster individually. This will provide a more accurate representation on the intra-cluster relationships.

The Sammon Mapping configuration shown in Figures 2.7(a) and 2.8 does not demonstrate any clear clusters, however a Classical Scaling output on the same data does suggest some cluster like features may exist. Figure 2.9 shows the Classical Scaling result with a possible cluster identified.

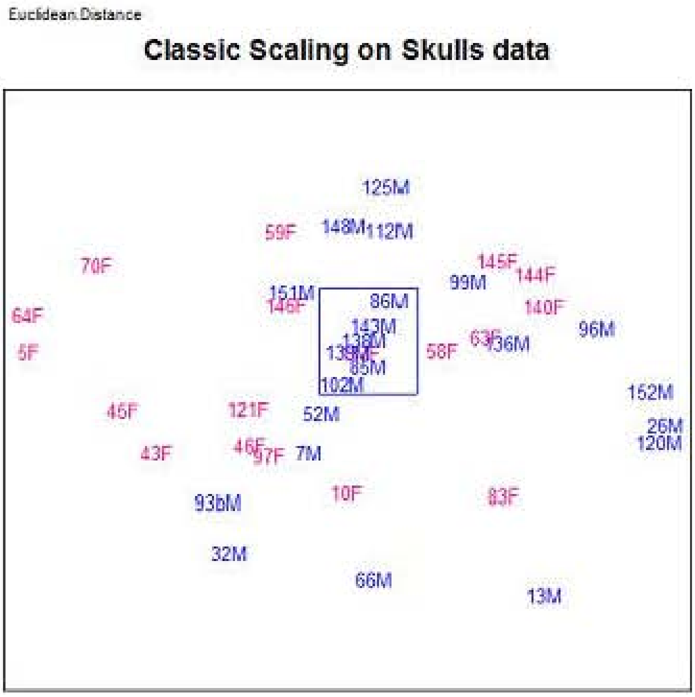


Figure 2.9: Skull Data: Cluster

In the event that $p > 3$ and consequently the final configuration is in a space that cannot be visualised, the researcher is usually required to perform analyses based on the numerical coordinates. An option however does exist to analyse separate components of the configuration in a visual manner, and attempt to make conclusions by combining the different visuals. So if $p = 4$, there is the option to plot dimension 1 vs. dimension 2, dimension 3 vs. dimension 4, etc. In order to perform this analysis in a comprehensive manner with two dimensional mappings a total of $\binom{p}{2}$ mappings exist. Similarly, if this were to be done using three dimensions, $\binom{p}{3}$ possible mappings exist.

2.10.2 Interpreting Axes and Dimension

Section 2.9 made reference to what are called 'axes of variation' in the scope of interpretation of MDS results. The example given was of in the case of breeds of dog being analysed and a possible axis of variation being aggressiveness. These axes should not be confused with the scaffolding axes in which the configuration is plotted, which will be equal to p. In fact the axes of variation are usually defined exclusively by the m variables in the original data set. The presence of these axes is therefore only applicable when the proximity data is based on a $n \times m$ matrix $\mathbf{Z}$ and not from a predefined $\mathbf{D}$ matrix. These m possible axes are defined in terms of the points and are thus affected by the orientation of configuration. They are defined with the use of a multiple linear regression model which is explained by Kruskal and Wish (1978) and elaborated on by Cox and Cox (2001). The regression prediction biplot axes method is used to construct these axes. For more information see Gower and Hand (1996), Section 3.3.2. The regression model has the dependent vector variable k as the k^{th} column from the data $\mathbf{Z}$. The independent variables are the coordinates of the points in the final configuration. These coordinates are in the form of matrix $\mathbf{X}$. The regression model for original variable k is then:

$$\mathbf{z}_k = \mathbf{X}\boldsymbol{\beta}_k + \boldsymbol{\epsilon}$$

An estimate of $\boldsymbol{\beta}_k$, the parameter vector, is defined using the the least squares estimate given by:

$$\widehat{\boldsymbol{\beta}}_k = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{z}_k$$

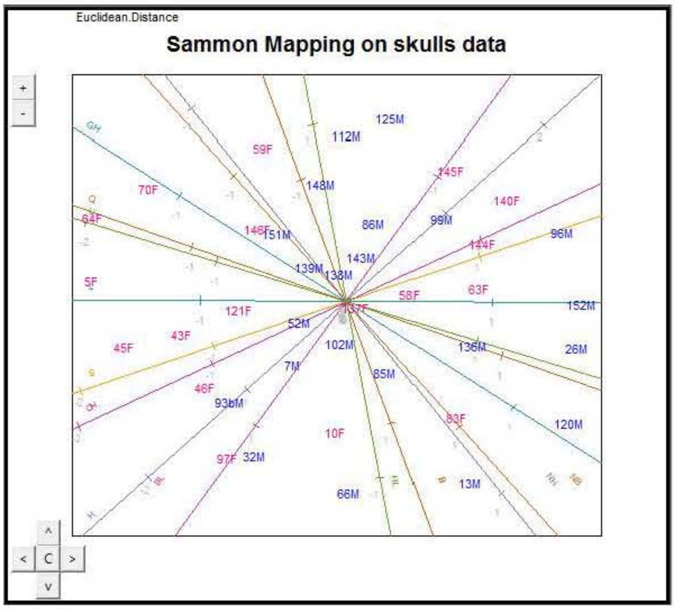
These axes are then defined to pass through the origin of the configuration based on the directional cosines, $\widehat{\boldsymbol{\beta}}_k/\sqrt{\sum \widehat{\boldsymbol{\beta}}_{ik}^2}$, and calibrated in the original units of measurement.

If each of the m axes are illustrated on the configuration, the plot will have m lines passing through the origin in different directions, with each line representing a variable from the data. The visualisation of these axes of variation consequently allow the observation of how each object in the configuration is influenced by each variable by assessing its orthogonal projection onto the axis. These axes have positive and negative ends, and the correlation of variables can be assessed according to how they run in comparison to each another. So, for the breeds of dog example in Section 2.9, objects (breeds) positioned towards the positive end of the axis (axis of aggression) will indicate more aggressive breeds, while breeds positioned towards the other end indicate less aggressive breeds. The skulls data set that has been used thus far in the Section consists of twelve variables, which are described in the Appendix. This means that there are twelve axes of variation that may be viewed. Figure 2.10(a) illustrates all twelve of these axes being displayed on the Sammon Mapping example (Figure 2.7(a)) and Figure 2.10(b) just a single axis for individual interpretation.

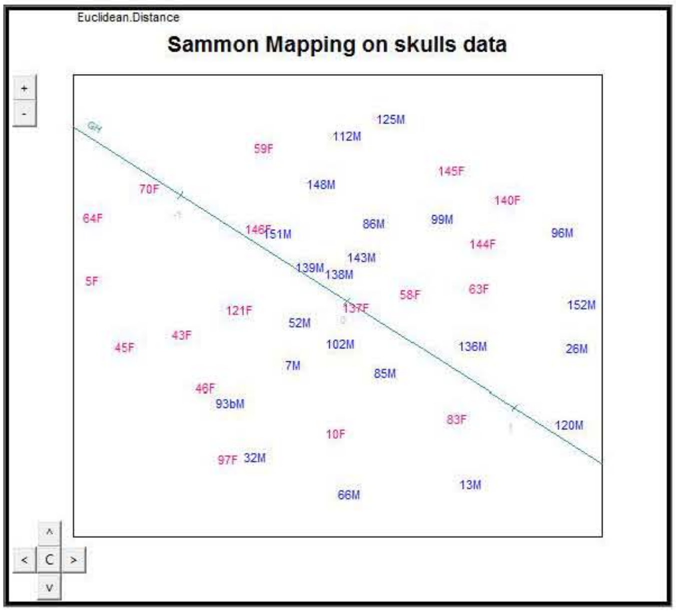
Figure 2.10(b) displays the axis of variation “G’H”, which refers to the ‘Upper Face Height’ measurement of a skull. By observing where points lie in relation to this axis, it can easily be seen that Male subjects tend towards the ‘positive’ end Female subjects tend to the negative.

2.10.3 Comparing Configurations

Since Multidimensional Scaling has so many forms, in application it is usual to produce a number of different configurations on the same set of data throughout a Section of research. For example, Figures 2.8 and 2.9 show different two dimensional configurations from the same data set using Sammon Mapping and Classical Scaling respectively. In these cases, it is often desirable to compare the two configurations directly in order to determine the extent and nature of the similarities and dissimilarities between them. A useful mathematical tool for making such comparisons is called ‘Orthogonal Procrustes Analysis’. The aim of Procrustes Analysis is to take two configurations, whose coordinate matrices are $n \times q$ and $n \times p$ respectively and obtain a one-to-one mapping from one set of points to the other in Euclidean Space. This is usually done in two dimensions, but theoretically could be done on a q dimensional space where $q < p$. Orthogonal Procrustes Analysis requires one the configurations to undergo a certain amount of manipulation. Forms of manipulation include translation, rotation and reflection, all of which in no way effect the distances between points in the configuration. Dilation is also used to manipulate the configuration. While dilation does



(a) Multiple Axes



(b) Single Axis

Figure 2.10: Skull Data: Axes of Variation

distort the actual distances between points, the ratio of distances remains unchanged and thus interpretation of relative relationships is still possible.

The result of this manipulation is that the configuration can be plotted over the other in such a way that the corresponding points from the separate sets are lined up with each other as closely as possible. Inspection of this mapping thus allows the researcher to observe the nature of any dissimilarities. The algebra behind such a process is quite complicated, however a practical summary of the steps of the procedure are provided by Cox and Cox (2000). The four step algorithm given below describes the process of matching configuration $\mathbf{Y}{:}n{\times}p$ to configuration $\mathbf{X}{:}n{\times}q$ when $p = q$: (The case where $p \neq q$ requires some level of projection. The reader is referred to the literature of Cox and Cox for detail on this alternative.)

1. Subtract the mean vectors for the configurations from each of the respective points in order to have the centroids at the origin.

2. Find the rotation matrix $\mathbf{A} = (\mathbf{X}^T\mathbf{Y}\mathbf{Y}^T\mathbf{X})^{\frac{1}{2}}(\mathbf{Y}^T\mathbf{X})^{-1}$ and rotate the $\mathbf{X}$ configuration to $\mathbf{XA}$

3. Scale the $\mathbf{X}$ configuration by multiplying each coordinate by ρ, where $\rho = tr(\mathbf{X}^T\mathbf{Y}\mathbf{Y}^T\mathbf{X})^{\frac{1}{2}}/tr(\mathbf{X}^T\mathbf{X})$

4. Calculate the minimised and scaled value of $R^2 = 1 - \{tr(\mathbf{X}^T\mathbf{Y}\mathbf{Y}^T\mathbf{X})^{\frac{1}{2}}\}^2/\{tr(\mathbf{X}^T\mathbf{X})tr(\mathbf{Y}^T\mathbf{Y})\}$ Where R^2 assesses the match of the two configurations and is referred to as the 'Procrustes Statistic'.

In Figure 2.11 a Procrustes Analysis is performed on the Sammon Mapping and Classical Scaling configurations of Figures 2.8 and 2.9. The output demonstrates the result where the points corresponding to the Sammon Mapping are in purple and those corresponding to the Classical Scaling are green. Inspection of this combined plot suggests that both methods have produced similar results in that all of the points have the same positioning. The Sammon Mapping, however, appears to have given greater spread in the axes, that is, in this case, vertical. Since the STRESS1 value associated with the Sammon Mapping and Classical Scaling is roughly 0.18 and 0.4 respectively, it can safely be determined that the greater vertical spread has contributed to a more accurate fit.

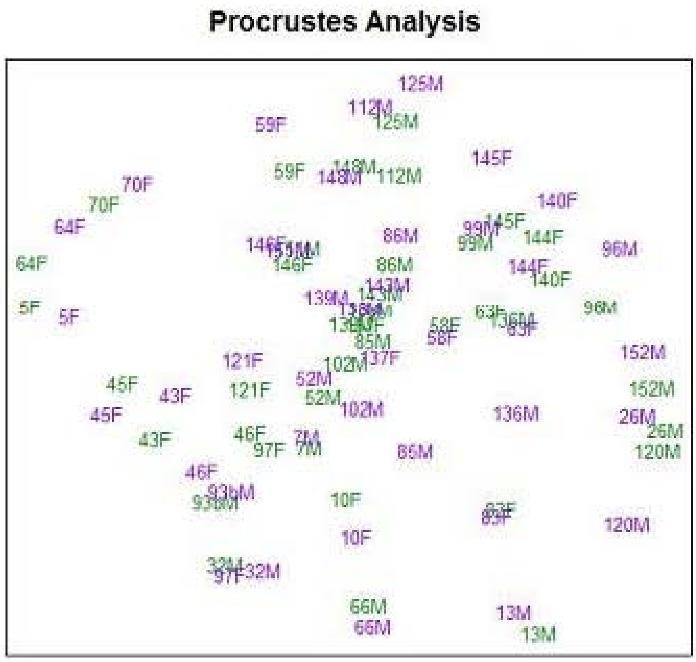


Figure 2.11: Skull Data: Procrustes Analysis Example

2.11 Applications for MDS

A strength of Multidimensional Scaling is in its versatility with regards to fields of application. Virtually any study that has a set of subjects that need to be compared will be suited to Multidimensional Scaling in a number of different formats. Torgerson (1952) was responsible for the naming of MDS and proposed the first application of it in the field of psychometrics. This particular first study used MDS to understand people's judgments of the similarity of members of a set of objects (Young, 1985).

The fields that have come to use MDS most commonly are Social Sciences, Marketing, Biometrics and Ecology. Some individual studies that have used MDS and earned some form a notoriety include: Lapointe and Lengendre (1994) study of whiskey, which is the basis of the example used in Section 2.7.2; The popular Morse-Code study by Rothkopf (1957); and Corradino (1990) study of primates.

2.12 Multivariate Methods Related to MDS

Multidimensional Scaling is not the only multivariate method that aims to portray information in many dimensions in a smaller space. Three related methods will be discussed briefly here, namely Principal Components, Correspondence Analysis and Cluster Analysis. Their similarities and dissimilarities with Multidimensional Scaling will also be addressed. The reader is referred to Borg and Groenen (2005) Chapter 24 for a detailed comparison between these methods and MDS.

2.12.1 Principal Component Analysis

The general idea of Principal Component Analysis (PCA), which was conceptualised and developed by Pearson (1901) and then Hotelling (1936), is in reducing a data matrix of many variables to a matrix of fewer variables. Consider the matrix $\mathbf{Z}$:$n \times m$ (n objects and m variables). Principal Component Analysis seeks to find a new set of p variables that are linear combinations of the original variables, where $p << m$ (Borg and Groenen, 2005). Therefore $\mathbf{X}_i = w_1(\mathbf{Z}_1) + w_2(\mathbf{Z}_2) + ... + w_m(\mathbf{Z}_m)$, where the w_j are unknown. Graphical depiction of the Principal Component result depends on the number of p variables used. Plotting the first two principal components gives a two dimensional configuration, the first three gives a three dimensional configuration, and so on.

Interpretation of configurations produced by PCA is much the same as those by MDS, in that relative similarity of points may be observed. One of the major differences in the procedures however is in the input data, seeing that by definition PCA must use an object by variable $\mathbf{Z}$ matrix, whereas MDS ordinarily uses the $n \times n$ proximity matrix $\mathbf{\Delta}$. A consequence of this is that configurations of PCA do not have diagnostic tools such as the Shepard Diagram at its disposal when analysing the results.

A method very closely related to PCA is Principal Coordinate Analysis (PCO). This method emphasises the representation of the objects (Gower, 1966). As already mentioned, PCO is essentially Classical Scaling. Details of which can be found in Chapter 3. Gower et al. (2011) shows that PCO based on a $\mathbf{\Delta}$ matrix of Euclidean distances between the rows of $\mathbf{Z}$ is equivalent to a PCA of $\mathbf{Z}$.

2.12.2 Correspondence Analysis

Correspondence Analysis (CA), first conceptualised by Hirschfeld (1935), employs methods similar to PCA but focused for the use of contingency tables with two categorical variables. In general, CA derives orthogonal factors from the data using the Chi-Squared statistic (Greenacre, 2007). Rows and columns of the contingency table are treated equivalently.

Similar to MDS, CA is used to graphically display objects as points in a low dimensional space, however MDS is usually used on one-mode data whereas CA is a two mode technique (and therefore is most comparable to the MDS method known as *Unfolding* which is not discussed in this dissertation). The data for CA is exclusively non-negative while MDS can construct $\mathbf{\Delta}$ even from negative data. CA is also limited to using the χ^2-distance as a dissimilarity measure, whereas MDS accepts a wide range of metrics in the derivation of $\mathbf{\Delta}$.

For further details on the differences between CA and MDS, the reader is referred to Heiser and Meulman (1983). For an assessment on the close relationship between CA and Classical Scaling, refer to Borg and Groenen (2005).

2.12.3 Cluster Analysis

Another group of methods for defining data by the groups that lie within it is collectively known as Cluster Analysis. Like Multidimensional Scaling, Cluster analysis consists of a number of methods, each with different algorithms and applications. Most Cluster Analysis methods can be classified as either hierarchical or non-hierarchical and some algorithms for cluster definition include: single linkage, complete linkage, average linkage, Ward's distance and the centroid method (Sambamoorth, 2012).

In most Cluster Analyses, the result is simply just an indication of which objects belong with which cluster where each object in the cluster is considered to belong to it equally. Cluster Analysis can therefore be used in conjunction with other methods such as Multidimensional Scaling. Using the result of a Cluster Analysis, a categorical variable may be added to the data to be used in MDS, indicating in which cluster each object was found to belong. The MDS result will then distinguish between the clusters in the layout and they may be analysed in greater detail.

2.13 Drawbacks of MDS

While Multidimensional Scaling has proven its use over decades and is a valuable tool for many statisticians, it does have some inevitable disadvantages. Two of these are the computational intensity of the methods and the second is the problem of local minima. Each of these will be discussed here.

MDS can be found to be computationally time consuming. Since the number of elements in the dissimilarity matrix $\mathbf{\Delta}$ and $\mathbf{D}$ grow exponentially with the increase in objects n, unsurprisingly, very large data sets require a greater allocation of computational resources than smaller data sets and thus are found to take even more time. The reason that MDS can demand so much computation is due to its iterative nature. If data sets are large and tolerance is small, iterations can extend into the order of 100's, each of which require complicated computation on large matrices. The improvement of modern computing power is changing this aspect of data analysis and making it less of a problem, however even standard desktops of today may take longer than expected when data is of a certain high magnitude. With regards to MDS software packages, such as the MDS-GUI, the problem of computation power applies when the iterative nature of the procedure is graphically displayed. The user of the software may be required to choose between speed and visual quality if the particular data set is too big.

The second drawback to Multidimensional Scaling is far more a comment on the technique itself than on technology, and this is the problem of local minima. Even users who have been experimenting with MDS software for a short time will observe that the final configuration can differ drastically when adjusting the parameters (while keeping the input proximities, the method of MDS and p constant). The iterations within the procedure will cease when the difference in stress values between two consecutive iterations is below the threshold value referred to as 'tolerance' which indicates some form of minimum has been achieved. Ideally, this minimum will be the global minimum and thus the configuration will be as good as possible. However, it is often the case that these minima are local and in these cases the configuration is not ideal. The major problem with this is that there is practically no indication of the extent of the difference between the global and local minima, which implies the researcher must experiment with parameters to try gain some understanding of the minima through trial and error. If a data set proves to have a problem with different points of convergence, the tolerance should be made more strict (decreased). A very low tolerance will make the definition of a minima more strict and therefore decrease the number of potential minima. The starting configuration can have a huge effect on the final configuration. For this reason, another suggestion for avoiding local minima is using starting configurations that have already proven to be accurate.

Chapter 3

Mathematics of Multidimensional Scaling

Chapter 2 gave an overview of Multidimensional Scaling from a theoretical perspective and explained the general concepts behind the range of ordination methods. Chapter 3 will be focused on the mathematical perspective of the multivariate techniques. Multidimensional Scaling can be divided into two main categories, being Metric and Non-Metric. The second and third Sections of this Chapter will therefore correspond to these two categories, where each will describe some of the key MDS techniques of that category. Finally a noteworthy independent method of MDS will be discussed.

3.1 General Mathematical Results

Before embarking on a discussion of the different MDS techniques a general mathematical result used throughout this Chapter will be given.

3.1.1 Differentiation of Euclidean Distances

The elements of the matrix $\mathbf{D}$ is given by

$$d_{rs} = \left[\sum_{i=1}^{p} (x_{ri} - x_{si})^2 \right]^{\frac{1}{2}}$$

The following cases need consideration:

$$\frac{\partial d_{rs}}{\partial x_{rk}} = \frac{1}{2}\left[\sum_{i=1}^{p}(x_{ri}-x_{si})^2\right]^{-\frac{1}{2}}(2)(x_{rk}-x_{sk}) = \frac{1}{d_{rs}}(x_{rk}-x_{sk})$$

$$\frac{\partial d_{rs}}{\partial x_{sk}} = \frac{1}{2}\left[\sum_{i=1}^{p}(x_{ri}-x_{si})^2\right]^{-\frac{1}{2}}(2)(x_{rk}-x_{sk})(-1) = \frac{-1}{d_{rs}}(x_{rk}-x_{sk})$$

$$\frac{\partial d_{rs}}{\partial x_{tk}} = 0 \quad if\ \ t \neq r, s$$

Define the indicator function $I^{rs} = \begin{cases} 1, & \text{if } r = s. \\ 0, & \text{otherwise.} \end{cases}$

Now it can be written that

$$\frac{\partial d_{rs}}{\partial x_{tk}} = \frac{1}{d_{rs}}(x_{rk}-x_{sk})(I^{rt}-I^{st}) \tag{3.1}$$

for $t = 1, ..., n$ and $k = 1, ..., p$.

3.2 Metric MDS

Section 2.2.1 gave a brief description of the idea behind Metric Multidimensional Scaling. The techniques of Metric Multidimensional Scaling are structured in such a way that there is an assumption of metric qualities in the measurement of the proximities. That is to say that the extent of the (dis)similarities between points are taken into account. Thus the distances in metric MDS space preserve the intervals and ratios as well as possible (Wickelmaier, 2003). The use of Metric Multidimensional Scaling is only valid when the assumption of metric distances can be justified. Continuing with the notation defined in Chapter 2, all methods of Metric Multidimensional Scaling aim to do the following (Cox and Cox, 2001):

With n objects and the subsequent proximities δ_{rs}, Metric MDS attempts to find a set of points in a p dimensional space such that the distances between these points, d_{rs} resemble as closely as possible δ_{rs}. Formally this is done in such a way that

$$d_{rs} \approx f(\delta_{rs})$$

where $f(\delta_{rs})$ is a continuous parametric monotonic function. This function is commonly in the form of the identity function, or alternatively attempts to transform the proximities to a distance like form. For ease of interpretation and explanation, it will be assumed that all proximities are already in a satisfactory distance like form and so the former will be used. That is to say, Metric MDS will be assumed to attempt to find d_{rs} such that $d_{rs} \approx \delta_{rs}$. The goodness-of-fit of the resulting $\mathbf{X}$ matrix, from which d_{rs} was derived, can be assessed using any of the *stress* formulas provided in Section 2.4. It should be noted that other variations of loss function are often used, as will be demonstrated with Least Squares Scaling in Section 3.2.2. Since stress is being calculated from a Metric perspective, the $f(x_{ij})$ component in equation 2.1 is simply equal to δ_{ij}. The stress values are thus based on the sum of squares computation $\sum\sum(\delta_{ij} - d_{ij})^2$. Alternatively, in the case of Classical Scaling, the *Strain* measurement in Section 2.4.5 may be used.

3.2.1 Classical Scaling

The concepts of Classical Scaling was originally developed by Young and Householder (1938) but only elaborated on in a scaling sense in Torgerson (1952). Classical Scaling is very closely related to Principal Coordinate Analysis due to a reliance on analysis of eigenvalues. Details on this connection is provided in Section 2.12, where the relationships between MDS and other ordination methods are discussed. Unlike all other MDS techniques, Classical Scaling is usually a non-iterative process. The process of Classical Scaling is discussed in this Section in a somewhat simplified format. For a full description of the method which has all intermediary steps included, the reader is referred to Section 2.2.1 of *Multidimensional Scaling: Second Edition* by Cox and Cox (2001).

Classical Scaling aims to find the coordinate matrix $\mathbf{X}$ based on a eigenvalue decomposition of the scaler product $\mathbf{B} = \mathbf{X}\mathbf{X}^T$. The $\mathbf{B}$ term refers to an inner product matrix and is found in the following manner:

It is defined that the Euclidean distances between points r and s in $\mathbf{X}$ are given by (3.1) since $\mathbf{X}$ is in a p dimensional Euclidean Space.

$$\begin{aligned} {d^2}_{rs} &= (\mathbf{x}_r - \mathbf{x}_s)^T(\mathbf{x}_r - \mathbf{x}_s) \\ &= \mathbf{x}_r^T\mathbf{x}_r + \mathbf{x}_s^T\mathbf{x}_s + 2\mathbf{x}_r^T\mathbf{x}_s \end{aligned} \tag{3.2}$$

The elements of **B** are then defined as $b_{rs} = {x_r}^T x_s$. Also, since it is desirable to place the configuration in such a way that the centroid is at the origin, $\sum_{r=1}^{n} x_{ri} = 0$. Following this, it can be shown that

$$b_{rs} = -\frac{1}{2}\left(d_{rs}^2 - \frac{1}{n}\sum_{r=1}^{n} d_{rs}^2 - \frac{1}{n}\sum_{s=1}^{n} d_{rs}^2 + \frac{1}{n^2}\sum_{r=1}^{n}\sum_{s=1}^{n} d_{rs}^2\right) \tag{3.3}$$

It thus follows that

$$\mathbf{B} = \mathbf{HAH} \tag{3.4}$$

where **A** is defined as $-\frac{1}{2}d_{rs}^2$ from (3.3) and **H** is a centering matrix

$$\mathbf{H} = \mathbf{I} - n^{-1}\mathbf{1}\mathbf{1}^T \tag{3.5}$$

with **1** being the vector of 1's. The double centered **B** is then written in terms of its spectral decomposition

$$\mathbf{B} = \mathbf{V}\Lambda\mathbf{V}^T$$

where Λ is the diagonal matrix of eigenvalues of **B** and **V** is the matrix of corresponding normalised eigenvectors. The number of p^* positive eigenvalues are then ordered in the convenient manner, such that $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq ...\lambda_{p^*} \geq 0$. And hence, **B** can be re-written as

$$\mathbf{B}_1 = \mathbf{V}_1\Lambda_1\mathbf{V}_1^T$$

with $\Lambda_1 = diag(\lambda_1...\lambda_{p^*})$ and $\mathbf{V}_1 = [v_1, ..., v_{p^*}]$. Since $\mathbf{B} = \mathbf{X}\mathbf{X}^T$, **X** can be found by (3.6)

$$\mathbf{X} = \mathbf{V}_1\Lambda_1^{\frac{1}{2}} \tag{3.6}$$

The appropriate coordinates for a desired p number of dimensions is thus the **X** corresponding to the p largest eigenvalues and associated eigenvectors. A useful feature of Classical Scaling is that of the nested dimensions property. This means that, unlike other forms of MDS, the first $p-1$ dimensions of a p dimensional result will correspond exactly to the result when a $p-1$ dimensional case is sought. A useful four step formula for performing Classical Scaling is given by Borg and Groenen (2005).

1. Compute the matrix of squared dissimilarities, $\mathbf{\Delta}^2$.

2. Apply double centering to this matrix. Thus $\mathbf{B} = \mathbf{HAH}$, where $\mathbf{A} = \left\{-\frac{1}{2}d^2\right\}$ and $\mathbf{H} = \mathbf{I} - n^{-1}\mathbf{1}\mathbf{1}^T$

3. Compute the eigendecomposition of $\mathbf{B}$, thus giving $\mathbf{B} = \mathbf{V}\Lambda\mathbf{V}^T$.

4. With the means of the resulting equation, $\mathbf{X} = \mathbf{V}_1\Lambda_1^{\frac{1}{2}}$, find the coordinate matrix $\mathbf{X}$ corresponding to the user's desired p.

3.2.1.1 Classical Scaling using Optimisation

The Classical Scaling method just described is a strictly algebraic and non iterative process and does not find a solution through optimising any loss function. Alternative Classical Scaling solutions however have been developed where Classical Scaling is viewed as a problem solved by optimisation. Carroll et al. (1976), de Leeuw and Heiser (1982) and Ter Braak (1992) discussed Classical Scaling with linear constraints imposed. Borg and Groenen (2005) describes this process.

The solution $\mathbf{X}$:$n \times p$ is found as a set of linear constraints on the input objects by variables matrix $\mathbf{Z}$:$n \times m$ such that

$$\mathbf{X} = \mathbf{ZC}$$

where $\mathbf{C}$:$m \times p$ are weights to be optimised.

The minimising strain function is given by

$$\begin{aligned} S &= tr[(\mathbf{B} - \mathbf{XX}^T)(\mathbf{B} - \mathbf{XX}^T)^T] \\ &= tr[(\mathbf{B} - \mathbf{ZCC}^T\mathbf{Z}^T)(\mathbf{B} - \mathbf{ZCC}^T\mathbf{Z}^T)^T] \end{aligned} \tag{3.7}$$

Let the singular value decomposition of $\mathbf{Z} = \mathbf{P\Phi Q}^T$. Then $\mathbf{X} = \mathbf{P\Phi Q}^T\mathbf{C} = \mathbf{PD}$ where $\mathbf{P}$:$n \times m$ has mutually orthonormal columns, $\mathbf{P}^T\mathbf{P} = \mathbf{I}_m$, and $\mathbf{D} = \mathbf{\Phi Q}^T\mathbf{C}$.

Now

$$\begin{aligned}
S &= tr[(\mathbf{B}-\mathbf{PDD}^T\mathbf{P}^T)(\mathbf{B}-\mathbf{PDD}^T\mathbf{P}^T)^T] \\
&= tr[\mathbf{BB}-\mathbf{BPDD}^T\mathbf{P}^T-\mathbf{PDD}^T\mathbf{P}^T\mathbf{B}+\mathbf{PDD}^T\mathbf{P}^T\mathbf{PDD}^T\mathbf{P}^T] \\
&= tr[\mathbf{BB}+\mathbf{PDD}^T\mathbf{DD}^T\mathbf{P}^T-2\mathbf{BPDD}^T\mathbf{P}^T] \\
&= tr[\mathbf{BB}+\mathbf{DD}^T\mathbf{DD}^T-2\mathbf{P}^T\mathbf{BPDD}^T+\mathbf{P}^T\mathbf{BPP}^T\mathbf{BP}-\mathbf{P}^T\mathbf{BPP}^T\mathbf{BP}] \\
&= tr[\mathbf{BB}^T-(\mathbf{P}^T\mathbf{BP})(\mathbf{P}^T\mathbf{BP})+(\mathbf{P}^T\mathbf{BP}-\mathbf{DD}^T)(\mathbf{P}^T\mathbf{BP}-\mathbf{DD}^T)] \\
&= tr(\mathbf{BB}^T)-tr[(\mathbf{P}^T\mathbf{BP})(\mathbf{P}^T\mathbf{BP})^T]+tr[(\mathbf{P}^T\mathbf{BP}-\mathbf{DD}^T)(\mathbf{P}^T\mathbf{BP}-\mathbf{DD}^T)^T]
\end{aligned} \tag{3.8}$$

Only the final term in (3.8) depends on $\mathbf{D}$. (3.8) will be a minimum when

$$\mathbf{P}^T\mathbf{BP} = \mathbf{DD}^T$$

i.e. $\mathbf{P}^T\mathbf{V\Lambda V}^T\mathbf{P} = \mathbf{DD}^T = (\mathbf{P}^T\mathbf{V\Lambda}^{\frac{1}{2}})(\mathbf{P}^T\mathbf{V\Lambda}^{\frac{1}{2}})^T$

Now $\mathbf{P}^T\mathbf{V\Lambda}^{\frac{1}{2}}$ is an $m \times n$ matrix and $\mathbf{D}$ is $m \times p$. It follows from the Eckart and Young theorem (Eckart and Young, 1936) that (3.8) is a minimum when $\mathbf{D} = \mathbf{P}^T\mathbf{V}_1\mathbf{\Lambda}_1^{\frac{1}{2}}$ with $\mathbf{V}_1$ and $\mathbf{\Lambda}_1^{\frac{1}{2}}$ the first p columns as defined in (3.6)

The weight matrix $\mathbf{C}$ is then given by solving

$$\mathbf{\Phi Q}^T\mathbf{C} = \mathbf{P}^T\mathbf{V}_1\mathbf{\Lambda}_1^{\frac{1}{2}}$$

yielding $\mathbf{C} = \mathbf{Q\Phi}^{-1}\mathbf{P}^T\mathbf{V}_1\mathbf{\Lambda}_1^{\frac{1}{2}}$.

The process repeats until the difference in loss statistics between iterations is negligible and thus the optimised $\mathbf{C}$ has been found.

For an alternative method of Classical Scaling with optimisation, see Lewis and Trosset (2009) who show a method of variable alternation and non-convex duality. It should be noted that the MDS-GUI will only supply the Classical Scaling method described in 3.2.1.

3.2.2 Least Squares Scaling

Metric Least Squares Scaling attempts to approximate δ_{rs} with d_{rs} by means of minimisation of a loss function. The method was first suggested by Sammon (1969) and can be seen as a Metric form of the Non-Metric Sammon

Mapping which will be discussed in Section 3.3.2. Similar to the Classical Scaling above, the following is derived from Cox and Cox (2001) which should be referred to for further discussion on the topic. The minimising loss function is called S and is given in the following equation:

$$S = \frac{\sum_{r<s} \delta_{rs}^{-1}(d_{rs} - \delta_{rs})^2}{\sum_{r<s} \delta_{rs}} \tag{3.9}$$

The numerator of the equation uses δ_{rs}^{-1} as a weighting factor and has the effect of giving smaller dissimilarities in the configuration a greater weighting in the loss function. In addition the summed component, $\sum_{r<s} \delta_{rs}$ is used as a normalising term and serves the purpose of making S scale free.

Using the result of (3.1) the following is shown:

$$\begin{aligned}
\frac{\partial S}{\partial x_{tk}} &= \frac{2\sum_{r<s} \delta_{rs}^{-1}(d_{rs} - \delta_{rs})}{\sum_{r<s} \delta_{rs}} \frac{\partial d_{rs}}{\partial x_{tk}} \\
&= \left(\frac{2}{\sum_{r<s} \delta_{rs}}\right) \sum_{r<s} \frac{d_{rs} - \delta_{rs}}{\delta_{rs}} \frac{x_{rk} - x_{sk}}{d_{rs}} (I^{rt} - I^{st}) \\
&= \left(\frac{2}{\sum_{r<s} \delta_{rs}}\right) \left\{ \sum_{s=2}^{n} \frac{(d_{1s} - \delta_{1s})(x_{1k} - x_{sk})}{\delta_{1s} d_{1s}} (I^{1t} - I^{st}) \right. \\
&+ \sum_{s=3}^{n} \frac{(d_{2s} - \delta_{2s})(x_{2k} - x_{sk})}{\delta_{2s} d_{2s}} (I^{2t} - I^{st}) \\
&+ \ldots \\
&+ \sum_{s=n-1}^{n} \frac{(d_{n-2,s} - \delta_{n-2,s})(x_{n-2,k} - x_{sk})}{\delta_{n-2,s} d_{n-2,s}} (I^{n-2,t} - I^{st}) \\
&+ \left. \frac{(d_{n-1,n} - \delta_{n-1,n})(x_{n-1,k} - x_{sk})}{\delta_{n-1,n} d_{n-1,n}} (I^{n-1,t} - I^{st}) \right\}
\end{aligned} \tag{3.10}$$

but $(I^{rt} - I^{rs}) = 0$ if $t \neq r, s$ so that
if $t = 1$:

$$\frac{\partial S}{\partial x_{tk}} = \left(\frac{2}{\sum_{r<s} \delta_{rs}}\right) \left\{ \sum_{s=2}^{n} \frac{(d_{1s} - \delta 1s)(x_{1k} - x_{sk})}{\delta_{1s} d_{1s}} + 0 \right\}$$

if $t = 2:$

$$\frac{\partial S}{\partial x_{tk}} = \left(\frac{2}{\sum\limits_{r<s}\delta_{rs}}\right)\left\{\frac{(d_{12}-\delta 12)(x_{1k}-x_{2k})}{\delta_{12}d_{12}}(-1)+\sum_{s=3}^{n}\frac{(d_{2s}-\delta 3s)(x_{1k}-x_{sk})}{\delta_{2s}d_{2s}}+0\right\}$$

...

if $t = n-1:$

$$\begin{aligned}\frac{\partial S}{\partial x_{tk}} &= \left(\frac{2}{\sum\limits_{r<s}\delta_{rs}}\right)\left\{\frac{(d_{1,n-1}-\delta_{1,n-1})(x_{1k}-x_{n-1,k})}{\delta_{1,n-1}d_{1,n-1}}(-1)+...\right.\\ &+\frac{(d_{n-2,n-1}-\delta_{n-2,n-1})(x_{n-2,k}-x_{n-1,k})}{\delta_{n-2,n-1}d_{n-2,n-1}}(-1)\\ &\left.+\frac{(d_{n-1,n}-\delta_{n-1,n})(x_{n-1,k}-x_{n,k})}{\delta_{n-1,n}d_{n-1,n}}\right\}\end{aligned}$$

if $t = n:$

$$\begin{aligned}\frac{\partial S}{\partial x_{tk}} &= \left(\frac{2}{\sum\limits_{r<s}\delta_{rs}}\right)\left\{\frac{(d_{1,n-1}-\delta_{1,n-1})(x_{1k}-x_{n-1,k})}{\delta_{1,n-1}d_{1,n-1}}(-1)+...\right.\\ &+\frac{(d_{n-2,n-1}-\delta_{n-2,n-1})(x_{n-2,k}-x_{n-1,k})}{\delta_{n-2,n-1}d_{n-2,n-1}}(-1)\\ &\left.+\frac{(d_{n-1,n}-\delta_{n-1,n})(x_{n-1,k}-x_{n,k})}{\delta_{n-1,n}d_{n-1,n}}(-1)\right\}\end{aligned} \tag{3.11}$$

which can be simplified to

$$\frac{\partial S}{\partial x_{tk}} = \left(\frac{2}{\sum\limits_{r<s}\delta_{rs}}\right)\sum_{r=1}^{n}\frac{(d_{ts}-\delta_{ts})}{\delta_{ts}d_{ts}}(x_{tk}-x_{sk}) \tag{3.12}$$

In order to minimise the function, the $\frac{\partial S}{\partial x_{tk}} = 0$ must be solved with $t = 1, ..., n; k = 1, ..., p$. A number of methods have been suggested for this calculation, the most common of which is a method of steepest descent by Sammon. The following formula assigns $x_{tk}^{(m)}$ as the m^{th} iteration and therefore $x_{tk}^{(m+1)}$ is the iteration following it.

$$x_{tk}^{(m+1)} = x_{tk}^{(m)} - MF\frac{\partial S}{\partial x_{tk}} / \left|\frac{\partial^2 S}{\partial^2 x_{tk}}\right| \tag{3.13}$$

The *MF* component in (3.13) refers to what is called the "Magic Factor" and was named and conceptualised by Sammon. The Factor is usually 0.3 or 0.4 and its purpose is assisting in the convergence to the minimum of the function and thus decreasing the number of iterations in the process.

3.3 Non-Metric MDS

Non-Metric Multidimensional Scaling methods are first described in Section 2.2.2. In many situations the metric assumptions described above are too strong for the data at hand. It is under these circumstances that the use of Non-Metric Multidimensional Scaling techniques may be more appropriate. Under the theories of non-metric multidimensional scaling the extent of the proximities is irrelevant. Only the ordering of the (dis)similarities is factored during the derivation of the MDS configuration. This Section will provide a discussion of the theories of Non-Metric Multidimensional Scaling in general and then go on to specific methods of the category.

Non-Metric Multidimensional Scaling, while sharing the same objectives, differs from Metric MDS in subtle ways. The objective of Non-Metric techniques is still to find a configuration whose d_{rs} match as closely as possible the input proximities δ_{rs}. This too is represented with a configuration of n points in a p dimensional space. However, the primary difference that defines Non-Metric MDS is that since only the ordering of the objects is relevant to the calculations, this is all that is taken into account in determining the accuracy of a configuration. A monotonically increasing function is usually applied to the matrix of distances between points in order to achieve this. The resulting fitted values are then referred to as the $\hat{d}$ values, where

$$\hat{d}_{rs} = f(d_{rs})$$

such that $\hat{d}_{rs} \leq \hat{d}_{tu}$ whenever $\delta_{rs} \leq \delta_{tu}$. This ensures that the ordering of the original proximities are maintained. This concept can be demonstrated with the use of a simple example of isotonic regression (other transformation examples are listed in Section 2.7.3). Figure 3.1 below shows the first eight points on a hypothetical Shepard Plot from a Non-Metric MDS process. As described in Section 2.6.2, the Shepard Plot has original proximities on the

X-Axis while the Y-Axis simultaneously depicts the coordinate based d_{rs} and the fitted $\hat{d}_{rs}$.

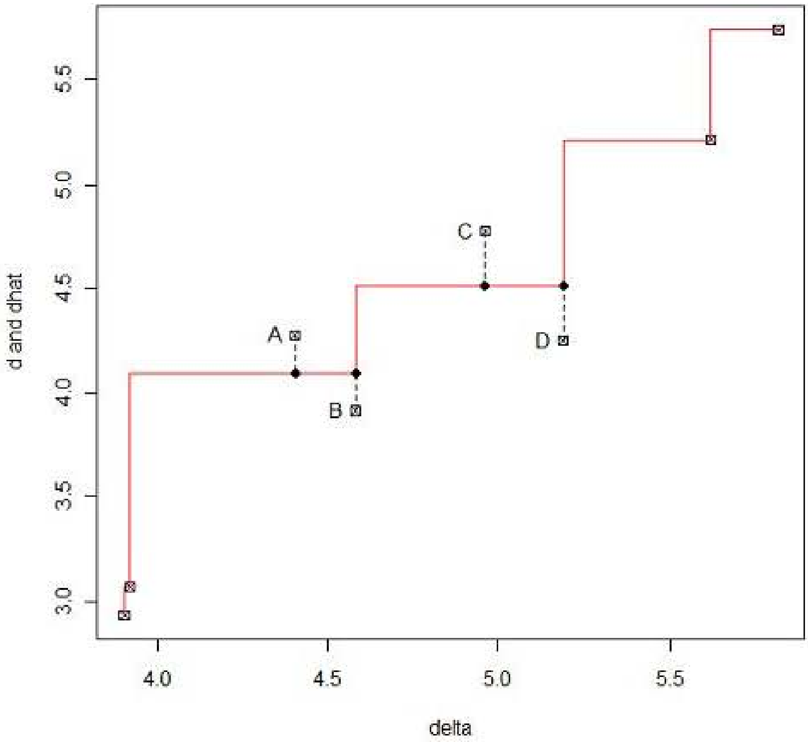


Figure 3.1: Non-Metric MDS: Transformation of Distances

Each point shown by a square in Figure 3.1 represents a pairing of points in the configuration with the Y-value corresponding to the MDS distance and X-value to the true observed proximity. The monotonic function is also displayed on the plot as the red step line. It is observed that all but four of the d_{rs} points are already fitted onto the line and thus $\hat{d}_{rs} = d_{rs}$. The points labeled A to D however lie off the function line. In these cases, the $\hat{d}$ values will be equal to the corresponding Y-values on the function line. This transformation is performed using the Up and Down Blocks Algorithm. The dotted lines relating to each of these points shows the transformation. Since the ordering of the deltas are unchanged, the Non-Metric requirement is satisfied. The goodness-of-fit of these configurations are still measured with the use of the Stress formulas as loss functions discussed in Section 2.4, where the iterative procedures attempt to minimise the Stress. However, unlike, Metric MDS, the $f(x_{ij}) = \hat{d}_{ij}$ and thus the sum of squares component is based on the differences between the distance of points in the configuration and their corresponding fitted values, thus $\sum \sum (\hat{d}_{ij} - d_{ij})^2$.

3.3.1 Kruskals MDS

The Non-Metric MDS method proposed by Kruskal (1964) makes use of STRESS-1 (Kruskal's Stress) as the loss function to be minimised in finding an accurate MDS based configuration. This was the first version of MDS to use a loss function of this nature. The transformation used in the method is specifically in the form of an isotonic regression which is nonparametric monotonely increasing. The reader is referred to Cox and Cox (2001) for a proof of the fact that

$$\sum_{i=1}^{N} (d_i - d_i^*)^2 \geq \sum_{i=1}^{N} (d_i - \hat{d}_i)^2$$

where $\{d_i^*\}$ is any arbitrary set of distances and $\{\hat{d}_i\}$ are the fitted values from the isotonic regression. The following discussion of Kruskal's method is also from Cox and Cox (2001).

For convenience, the loss function, in the form of STRESS-1, will be written in the following format

$$S = \sqrt{\frac{S^*}{T^*}} \tag{3.14}$$

with $S^* = \sum\limits_{r,s} (d_{rs} - \hat{d}_{rs})^2$ and $T^* = \sum\limits_{r,s} d_{rs}^2$.

From (3.14) it follows that $S^2 = S^* T^{*-1}$ and the differential equation follows as

$$\begin{aligned}
2S\frac{\partial S}{\partial x_{ui}} &= \frac{\partial S^*}{\partial x_{ui}}T^{*-1} + S^*(-T^{*-2})\frac{\partial T^*}{\partial x_{ui}} \\
\frac{\partial S}{\partial x_{ui}} &= \frac{1}{2S}\frac{1}{T^*}\left[\frac{\partial S^*}{\partial x_{ui}} - \frac{S^*}{T^*}\frac{\partial T^*}{\partial x_{ui}}\right] \\
&= \frac{1}{2}\frac{\sqrt{T^*}}{\sqrt{S^*}}\frac{S^*}{T^*}\left[\frac{1}{S^*}\frac{\partial S^*}{\partial x_{ui}} - \frac{1}{T^*}\frac{\partial T^*}{\partial x_{ui}}\right] \\
&= \frac{1}{2}S\left[\frac{1}{S^*}\frac{\partial S^*}{\partial x_{ui}} - \frac{1}{T^*}\frac{\partial T^*}{\partial x_{ui}}\right] \qquad (3.15)
\end{aligned}$$

where $\frac{\partial S^*}{\partial x_{ui}} = 2\sum_{r,s}(d_{rs} - \hat{d}_{rs})\frac{\partial d_{rs}}{\partial x_{ui}}$ and $\frac{\partial T^*}{\partial x_{ui}} = 2\sum_{rs} d_{rs}\frac{\partial d_{rs}}{\partial x_{ui}}$
Once again, using result (3.1), (3.15) becomes

$$\begin{aligned}
\frac{\partial S}{\partial x_{ui}} &= \frac{1}{2}S\left[\frac{2}{S^*}\sum_{r,s}(d_{rs} - \hat{d}_{rs})\frac{\partial d_{rs}}{\partial x_{ui}} - \frac{2}{T^*}\sum_{rs} d_{rs}\frac{\partial d_{rs}}{\partial x_{ui}}\right] \\
&= S\left[\sum_{r,s}\left\{\frac{d_{rs} - \hat{d}_{rs}}{S^*} - \frac{d_{rs}}{T^*}\right\}\frac{1}{d_{rs}}|x_{ri} - x_{si}|(I^{ru} - I^{su})\right] \\
&= S\sum_{r,s}(I^{ru} - I^{su})\left[\frac{d_{rs} - \hat{d}_{rs}}{S^*} - \frac{d_{rs}}{T^*}\right]\frac{|x_{ri} - x_{si}|}{d_{rs}} \qquad (3.16)
\end{aligned}$$

The minimum is then found using a method of steepest descent with the following iterative formula

$$\mathbf{X}_{m+1} = \mathbf{X}_m - \frac{\frac{\partial S}{\partial \mathbf{X}} \times sl}{|\frac{\partial S}{\partial \mathbf{X}}|} \qquad (3.17)$$

with $\mathbf{X}_m$ being the m^{th} configuration and the initial $\mathbf{X}$ being some starting configuration $\mathbf{X}_0$. The *sl* term in the formula refers to, what Kruskal described as, the 'step length'. This *step length* component is itself iterative in nature and has the following form.

$$sl_{present} = sl_{previous} \times (anglefactor) \times (relaxationfactor) \times (goodluckfactor)$$

With the angle factor $= 4.0^{cos^3\theta}$ where $\theta =$ the angle between the present and previous gradients ($\frac{\partial S}{\partial \mathbf{X}}$). The relaxation factor $\frac{1.3}{1+(5stepratio)^5}$ where the '5 step ratio' is $= min\left[1, \left(\frac{presentstress}{stress5iterationsago}\right)\right]$. Finally the 'good luck factor' is defined as the $min\left[1, \frac{presentstress}{previousstress}\right]$. This process can be summarised with the following step by step formula.

1. Choose $\mathbf{x}_0$
2. Find $\{d_{rs}\}$, the Euclidean distances between points in $\mathbf{X}$.
3. Fit the $\hat{d}_{rs}$ using an isotonic regression of the d_{rs} on δ_{rs}.
4. Find the gradient $\frac{\partial S}{\partial \mathbf{X}}$. If $\left|\frac{\partial S}{\partial \mathbf{X}}\right| < \epsilon$ where epsilon is a predefined tolerance value, the minimum stress has been achieved by the n^{th} configuration and the process can stop. If not, proceed to step 5.
5. Compute the 'step length', *sl.*
6. Find configuration *n+1* using (3.17)
7. Go to step 2.

3.3.2 Sammon Mapping

Sammon Mapping, also developed by Sammon (1969) is closely affiliated with Metric Least Squares Scaling, which is discussed in Section 3.2.2. Many texts refer to 'Sammon Mapping' as a Metric MDS process, and in these cases the MDS in question is precisely Metric Least Squares Scaling. This dissertation however will refer to Sammon Mapping in a Non-Metric MDS sense, which is in accordance with the *sammon* function of the **MASS** package. This function performs the primary calculations of Sammon Mapping within the applications of the MDS-GUI. Details of this function and package can be found in Section 4.2.2.2.

The Non-Metric loss function associated with Sammon Mapping can then be seen to be (3.18) which is clearly a Non-Metric version of (3.9).

$$S = \frac{\sum\limits_{r<s} \hat{d}_{rs}^{-1}(d_{rs} - \hat{d}_{rs})^2}{\sum\limits_{r<s} \hat{d}_{rs}} \tag{3.18}$$

Following the logic given in Section 3.2.2, (3.18) is minimised by the following.

$$\frac{\partial S}{\partial x_{tk}} = \left(\frac{2}{\sum\limits_{r<s} \hat{d}_{rs}} \right) \sum_{r=1}^{n} \frac{(d_{rt} - \hat{d}_{rt})}{\hat{d}_{rt} d_{rt}} (x_{rk} - x_{tk}) \tag{3.19}$$

This is solved by the steepest descent method first introduced in (3.13)

$$x_{tk}^{(m+1)} = x_{tk}^{(m)} - MF \frac{\partial S}{\partial x_{tk}} / \left| \frac{\partial^2 S}{\partial^2 x_{tk}} \right|$$

Once again, the *MF* component refers to the 'Magic Factor' developed by Sammon (1969).

3.4 SMACOF

The SMACOF methods of Multidimensional Scaling are considered by many to be one of the most accurate and convenient ways of performing MDS. The name is an acronym for "Scaling by Majorising a Complicated Function". As the name suggests, SMACOF techniques aim to attain the minimum of the appropriate loss function by means of a majorisation algorithm. SMACOF has both Metric and Non-Metric formats, both of which will be discussed in this Section. Firstly however, the majorisation algorithm must be summarised.

3.4.1 The Majorisation Algorithm

The use of majorisation as a minimisation technique is one that is ever increasing in popularity (Borg and Groenen, 2005). Borg and Groenen (2005) describe a feature of iterative majorisation to be generating a monotonically non increasing sequence of function values. The idea behind the method is that each of these generated auxiliary functions be less complicated than the original loss function to be minimised. This allows an iterative procedure of minimisation of the current auxiliary function, where minimisation of the simple function is less complicated. Figure 3.2 will aid the explanation of the process where the minimum of some function $f(x)$ must be located.

The function to be minimised, $f(x)$, is shown in purple of Figure 3.2. The majorisation process uses some appropriate auxiliary functions $g(x, y)$,

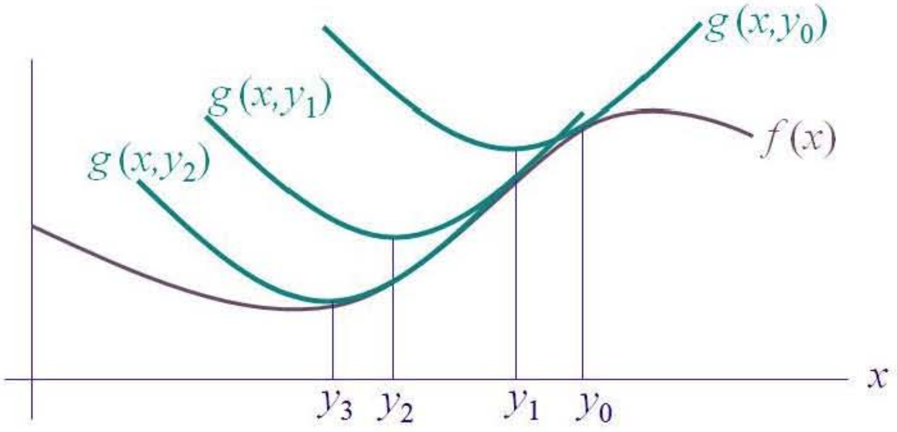


Figure 3.2: Majorising Algorithm Example

in green, where $f(y) = g(y, y)$. These functions must meet the following requirements (Borg and Groenen, 2005).

1. $g(x, y)$ must be more simple to minimise than $f(x)$.

2. $g(x, y)$ must always be greater than or equal to $f(x)$.

3. $g(x, y)$ must share exactly one point of tangency with $f(x)$. i.e. the functions must touch at one point with requirement 2 still holding.

The procedure requires a starting value for y, termed y_0, which describes the original location of the point of tangency between $f(x)$ and $g(x, y_0)$. From this point the algorithm proceeds as follows: The minimum value of $g(x, y_0)$ is found and the resulting y of this point is termed y_1 where by definition $y_1 \leq y_0$. The auxiliary function $g(x, y_1)$ is then defined where y_1 is at the point of tangency with $f(x)$. $g(x, y_1)$ will then be minimised to find y_2, and so on. The iterations will cease when $y_r - y_{r+1} < \epsilon$ at which point there is convergence and thus a minimum has been found.

The SMACOF methods of Multidimensional Scaling use these majorisation principals as a means of solving the loss function in order to find an optimal $\mathbf{X}$. Like many procedures dependent on optimisation, there is no real guarantee that the global minimum is reached and that the solution found is actually at a local minimum. Improvements on the optimisation technique, with specific SMACOF applications, have been developed, but are beyond the scope of this dissertation. For an example of one such method, refer to Groenen and Heiser (1996), where the Tunneling method for global optimisation is discussed.

3.4.2 Metric SMACOF

The following procedure is taken from Borg and Groenen (2005) and Cox and Cox (2001). The loss function to be minimised by SMACOF will be in the following form:

$$\begin{aligned} S &= \sum_{r<s}(\delta_{rs} - d_{rs})^2 \\ &= \sum_{r<s}\delta_{rs}^2 + \sum_{r<s} d_{rs}^2(\mathbf{X}) - 2\sum_{r<s}\delta_{rs}d_{rs}(\mathbf{X}) \end{aligned} \tag{3.20}$$

The first term in (3.20) is independent of $\mathbf{X}$. Next

$$\begin{aligned} \sum_{rs} d_{rs}^2(\mathbf{X}) &= \sum_{rs}\left\{\sum_i (x_{ri} - x_{si})^2\right\} \\ &= \sum_{rs} tr(\mathbf{X}^T\mathbf{A}_{rs}\mathbf{X}) = tr(\mathbf{X}^T\mathbf{A}\mathbf{X}) \end{aligned} \tag{3.21}$$

where the matrix $\mathbf{A}_{rs} : n \times n$ contains entries $a_{rr} = a_{ss} = 1$, $a_{rs} = a_{sr} = -1$ and all other entries zero. The matrix $\mathbf{A}$ has entries $a_{rs} = -1$ if $r \neq s$ and $a_{rr} = -\sum\limits_{1,r\neq s}^{n} a_{rs}$.

Furthermore

$\sum\limits_{r<s}\delta_{rs}d_{rs}(\mathbf{X}) = \sum\limits_{r<s}\delta_{rs}(\sum\limits_i (x_{ri} - x_{si})^2)^{\frac{1}{2}} = \sum\limits_{r<s}\delta_{rs}\sqrt{(\mathbf{x}_r - \mathbf{x}_s)^T(\mathbf{x}_r - \mathbf{x}_s)}$

The Cauchy-Schwarz inequality (Johnson and Wicheren, 2007) states that $(\mathbf{b}^T\mathbf{f})^2 \leq (\mathbf{b}^T\mathbf{b})(\mathbf{f}^T\mathbf{f})$ for any two equal length vectors $\mathbf{b}$ and $\mathbf{f}$ with equality if and only if $\mathbf{b} = c\mathbf{f}$ for some constant c.

Now

$$[(\mathbf{x}_r - \mathbf{x}_s)^T(\mathbf{y}_r - \mathbf{y}_s)]^2 \leq [(\mathbf{x}_r - \mathbf{x}_s)^T(\mathbf{x}_r - \mathbf{x}_s)][(\mathbf{y}_r - \mathbf{y}_s)^T(\mathbf{y}_r - \mathbf{y}_s)]$$

and

$$\begin{aligned}\sqrt{[(\mathbf{x}_r - \mathbf{x}_s)^T(\mathbf{x}_r - \mathbf{x}_s)]} &\geq \frac{(\mathbf{x}_r - \mathbf{x}_s)^T(\mathbf{y}_r - \mathbf{y}_s)}{\sqrt{(\mathbf{y}_r - \mathbf{y}_s)^T(\mathbf{y}_r - \mathbf{y}_s)}} \\ &= \frac{tr(\mathbf{X}^T\mathbf{A}_{rs}\mathbf{Y})}{d_{rs}(\mathbf{Y})}\end{aligned}$$

Therefore $\sum\limits_{r<s} \delta_{rs} d_{rs}(\mathbf{X}) \geq \sum\limits_{r<s} \frac{\delta_{rs}}{d_{rs}(\mathbf{Y})} tr(\mathbf{X}^T\mathbf{A}_{rs}\mathbf{Y})$

Define the matrix $\mathbf{B}(\mathbf{Y})$:$n \times n$ with elements

$$b_{rs} = \begin{cases} -\frac{\delta_{rs}}{d_{rs}(\mathbf{Y})}, & \text{for } r \neq s \text{ and } d_{rs}(\mathbf{Y}) \neq 0. \\ 0, & \text{for } r \neq s \text{ and } d_{rs}(\mathbf{Y}) = 0. \end{cases}$$

and $b_{rr} = -\sum\limits_{1,r\neq s}^{n} b_{rs}$

Then $-\sum\limits_{r<s} \delta_{rs} d_{rs}(\mathbf{X}) \leq -tr(\mathbf{X}^T\mathbf{B}(\mathbf{Y})\mathbf{Y})$

The minimum obtained by the Cauchy-Schwarz inequality is given by $\mathbf{X} = \mathbf{Y}$ so that

$$-\sum_{r<s} \delta_{rs} d_{rs}(\mathbf{X}) = -tr(\mathbf{X}^T\mathbf{B}(\mathbf{X})\mathbf{X}) \leq -tr(\mathbf{X}^T\mathbf{B}(\mathbf{Y})\mathbf{Y}) \tag{3.22}$$

Combining (3.21) and (3.22) the majorisation inequality is given by

$$\begin{aligned}S &= \sum_{r<s} \delta_{rs}^2 + tr(\mathbf{X}^T\mathbf{A}\mathbf{X}) - tr(\mathbf{X}^T\mathbf{B}(\mathbf{X})\mathbf{X}) \\ &\leq \sum_{r<s} \delta_{rs}^2 + tr(\mathbf{X}^T\mathbf{A}\mathbf{X}) - tr(\mathbf{X}^T\mathbf{B}(\mathbf{Y})\mathbf{Y})\end{aligned} \tag{3.23}$$

The minimum of the right hand side of (3.23) is obtained by setting the derivative equal to zero.

$$\begin{gathered}\frac{\partial}{\partial \mathbf{X}} = 2\mathbf{A}\mathbf{X} - 2\mathbf{B}(\mathbf{Y})\mathbf{Y} = 0 \\ \mathbf{A}\mathbf{X} = \mathbf{B}(\mathbf{Y})\mathbf{Y}\end{gathered} \tag{3.24}$$

Since the matrix $\mathbf{A}$ is not of full rank the system of equations is solved by the 'Guttman Transform' (Guttman, 1968) which in this case reduces to

$$\mathbf{X} = \frac{1}{n}\mathbf{B}(\mathbf{Y})\mathbf{Y}$$

The SMACOF algorithm guarantees a series of non-increasing Stress values. In the initial step set $\mathbf{Y} = \mathbf{X}^{(0)}$, the starting configuration. Then

$$\mathbf{X}^{(m+1)} = \frac{1}{n}\mathbf{B}(\mathbf{X}^{(m)})\mathbf{X}^{(m)}$$

The algorithm terminates when $S(\mathbf{X}^{(m)}) - S(\mathbf{X}^{(m+1)}) < \epsilon$ or the maximum number of iterations is reached.

3.4.3 Non-Metric SMACOF

The theory behind Non-Metric SMACOF is virtually identical with only a few subtle differences. Firstly, as expected the loss function S becomes.

$$S = \sum_{r<s}(\hat{d}_{rs} - d_{rs})^2$$

This change has the effect of the method requiring two separate minimisations. The first minimisation is respective of d_{rs} and can be done with majorisation as shown in Section 3.4.2. The second minimisation is respective of $\hat{d}_{rs}$ which may be performed using isotonic regression. At each iteration both minimisation steps are performed.

In general, the process goes as follows: From the starting configuration $\mathbf{X}_0$, the d_{rs} and the corresponding $\hat{d}_{rs}$ are calculated. S is then minimised with respect to d_{rs} and then to $\hat{d}_{rs}$, resulting in a new configuration $\mathbf{X}_1$. The process repeats until convergence.

3.5 Unidimensional Scaling

Unidimensional Scaling is the term used to describe Multidimensional Scaling in only one dimension. The graphical result of such a process is all n points on a single line, with the similarity or dissimilarity between ob9jects being observed by the distance between them on the line. Some authors, including Borg and Groenen (2005) and Cox and Cox (2001), consider Unidimensional Scaling to be an unreliable method. This is due to the fact that it is habitually prone to converging on local minima. Many MDS methods, including those already discussed, are able to make provisions for Unidimensional Scaling by simply setting $p = 1$. Unidimensional Scaling specific methods do exist, however, and the metric version of one of these will be discussed here.

The function to be minimised is given by (3.25) (Cox and Cox, 2000).

$$S = \sum_{r<s} (\delta_{rs} - |x_r - x_s|)^2 \tag{3.25}$$

where d_{rs} is represented by $|x_r - x_s|$, the distance between the points r and s on the straight line. Guttman (1968) shows that $\mathbf{x}$ is a local minimum of S if and only if

$$x_r = \frac{1}{n} \sum_{s=1}^{n} \delta_{rs} sign(x_r - x_s) \tag{3.26}$$

With $sign(x_r - x_s) = +$ when $x_r > x_s$ and $sign(x_r - x_s) = -$ when $x_r < x_s$. Guttman then goes on to propose the following algorithm for finding minima.

$$x_r^{(m+1)} = \frac{1}{n} \sum_{s=1}^{n} \delta_{rs} sign(x_r^{(m)} - x_s^{(m)}) \tag{3.27}$$

The $x_r^{(m)}$ in (3.27) is the value of x_r at the m^{th} iteration. In the event that $x_r^{(m)} = x_s^{(m)}$ and $r \neq s$, $sign(x_r - x_s)$ is then denoted by $+$ and the corresponding $sign(x_s - x_r)$ becomes $-$. The algorithm continues until $x_r^{(m+1)} - x_r^{(m+1)} \leq \epsilon$. Since the procedure is likely to be susceptible to many minima, a number of starting points are considered with the hope that the global minimum will be achieved.

Other approaches are suggested for performing Unidimensional Scaling, including methods using dynamic programming and methods using linear programming. The reader is referred to Cox and Cox (2000) for a summary of these.

3.6 INDSCAL

The following method of MDS is not at present available in the MDS-GUI. It is however intended to be the first method to be added in the next iteration of the software. Steps have already been made towards its inclusion, including a study of the method, which is described below.

INDSCAL was developed for types of data with more than two ways (refer to Section 2.7.2 for description of ways). The proximities associated with the data are thus denoted $\delta_{rs,i}$ where the i specifies the index of the set of observations, when N sets of observations exist. INDSCAL, being an acronym for 'Individual Differences Scaling', was developed by Carroll and Chang (1970).

The Metric INDSCAL method requires the derivation of two separate spaces, both with p dimensions. The first space, as usual, depicts the configuration of n points relating to the objects and is called 'The group stimulus space' (Cox and Cox, 2001). The second space, however, relates to the the N individuals ('The subject space'). These spaces are related in such a way that the coordinates of the individuals in the subject space are the weights in the weighted Euclidean distances between objects in the group stimulus space. This ensures that each individual's contribution to the proximities score is preserved. The weighted distance between points in the configuration is then defined by

$$d_{rs,i} = \left\{ \sum_{t=1}^{p} w_{it}(x_{rt} - x_{st})^2 \right\}^{\frac{1}{2}} \tag{3.28}$$

with w_{it} referring to the coordinate of the p^{th} dimension of the point relating to individual i from the subject space.

The algorithm for solving this problem proposed by Carroll and Chang (1970) proceeds as follows: The distances measured between points in the subject space are doubly centered producing matrix $\mathbf{B}_i$. Following the logic shown in Equations (3.2) and (3.3) in Section 3.2.1, it is shown that

$$\begin{aligned} b_{rs,i} &= \sum_{t=1}^{p} w_{it} x_{rt} x_{st} \\ &= \mathbf{H}\mathbf{A}_i\mathbf{H} \end{aligned} \tag{3.29}$$

with $\mathbf{H}$ once again defined as $\mathbf{H} = \mathbf{I} - n^{-1}\mathbf{1}\mathbf{1}^T$ and $[\mathbf{A}_i]_{rs} = -\frac{1}{2}d^2_{rs,i}$. S, the loss function is then used to minimise the least squares estimates of $\{w_{it}\}$

and $\{x_{rt}\}$.

$$\begin{aligned} S &= \sum_{r,s,i}\left(b_{rs,i} - \sum_{t=1}^{p} w_{it}x_{rt}x_{st}\right)^2 \\ &= \sum_{r,s,i}\left(b_{rs,i} - \sum_{t=1}^{p} w_{it}x_{rt}^{L}x_{st}^{R}\right)^2 = 0 \end{aligned} \tag{3.30}$$

The superscripts ‘L’ and ‘R’ are included (Left and Right) and are used to distinguish between two estimates of points in the group stimulus space which should converge to a common estimate. Minimising S requires three separate processes, being minimisation of $\{w_{it}\}$, $\{x_{rt}^{L}\}$ and $\{x_{st}^{R}\}$ respectively.

Consider N sets of observations

$$\mathbf{Z}_1 = \begin{bmatrix} z_{11,1} & z_{12,1} & \cdots & z_{1m,1} \\ \vdots & & & \\ z_{n1,1} & z_{n2,1} & \cdots & z_{nm,1} \end{bmatrix}, \ldots, \mathbf{Z}_N = \begin{bmatrix} z_{11,N} & z_{12,N} & \cdots & z_{1m,N} \\ \vdots & & & \\ z_{n1,N} & z_{n2,N} & \cdots & z_{nm,N} \end{bmatrix}$$

each yielding an $n \times n$ dissimilarity matrix

$$\mathbf{\Delta}_1 = \begin{bmatrix} 0 & \delta_{12,1} & \delta_{13,1} & \cdots & \delta_{1n,1} \\ \delta_{12,1} & 0 & \delta_{23,1} & \cdots & \delta_{2n,1} \\ \delta_{13,1} & \delta_{23,1} & 0 & \cdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \delta_{1n,1} & \delta_{2n,1} & \cdots & \cdots & 0 \end{bmatrix}, \ldots, \mathbf{\Delta}_N = \begin{bmatrix} 0 & \delta_{12,N} & \delta_{13,N} & \cdots & \delta_{1n,N} \\ \delta_{12,N} & 0 & \delta_{23,N} & \cdots & \delta_{2n,N} \\ \delta_{13,N} & \delta_{23,N} & 0 & \cdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \delta_{1n,N} & \delta_{2n,N} & \cdots & \cdots & 0 \end{bmatrix}$$

with double centered matrices $\mathbf{B}_1 = \mathbf{H}\mathbf{A}_1\mathbf{H}, \ldots, \mathbf{B}_N = \mathbf{H}\mathbf{A}_N\mathbf{H}$

and two matrices of initial configurations (which can be taken as the same configuration) of the subject space

$$\mathbf{X}^{L(0)} = \begin{bmatrix} x_{11}^{L(0)} & x_{12}^{L(0)} \\ \vdots & \vdots \\ x_{n1}^{L(0)} & x_{n2}^{L(0)} \end{bmatrix} \quad \text{and} \quad \mathbf{X}^{R(0)} = \begin{bmatrix} x_{11}^{R(0)} & x_{12}^{R(0)} \\ \vdots & \vdots \\ x_{n1}^{R(0)} & x_{n2}^{R(0)} \end{bmatrix}$$

Carroll and Chang (1970) uses a recursive least squares approach, minimising in turn $\{w_{it}\}$, $\{X^L_{rt}\}$ and $\{X^R_{rt}\}$ while keeping the other two sets fixed.

Define $\mathbf{F}$:$N \times n^2$ with $f_{ij} = b_{rs,i}$ and $j = n(r-1) + s$.

$$\mathbf{F} = \begin{bmatrix} b_{11,1} & b_{12,1} & \dots & b_{1n,1} & b_{21,1} & \dots b_{nn,1} \\ b_{11,2} & b_{12,2} & \dots & b_{1n,2} & b_{21,2} & \dots b_{nn,2} \\ \vdots & & & & & \\ b_{11,N} & b_{12,N} & \dots & b_{1n,N} & b_{21,N} & \dots b_{nn,N} \end{bmatrix}$$

and $\mathbf{G} : n^2 \times p$ with $g_{ij} = x^L_{rj} x^R_{sj}$ and $j = n(r-1) + s$.

$$\mathbf{G}^{(m)} = \begin{bmatrix} x^{L(m)}_{11} x^{R(m)}_{11} & x^{L(m)}_{12} x^{R(m)}_{12} \\ x^{L(m)}_{11} x^{R(m)}_{21} & x^{L(m)}_{12} x^{R(m)}_{22} \\ \vdots & \vdots \\ x^{L(m)}_{11} x^{R(m)}_{n1} & x^{L(m)}_{12} x^{R(m)}_{n2} \\ x^{L(m)}_{21} x^{R(m)}_{11} & x^{L(m)}_{22} x^{R(m)}_{12} \\ \vdots & \vdots \\ x^{L(m)}_{21} x^{R(m)}_{n1} & x^{L(m)}_{22} x^{R(m)}_{n2} \\ \vdots & \vdots \\ x^{L(m)}_{n1} x^{R(m)}_{11} & x^{L(m)}_{n2} x^{R(m)}_{12} \\ \vdots & \vdots \\ x^{L(m)}_{n1} x^{R(m)}_{n1} & x^{L(m)}_{n2} x^{R(m)}_{n2} \end{bmatrix} \tag{3.31}$$

The least squares estimate of $\mathbf{W} : N \times p$ is given by $\mathbf{W}^{(m+1)} = \mathbf{F}\mathbf{G}^{(m)}(\mathbf{G}^{(m)T}\mathbf{G}^{(m)})^{-1}$

With the coordinate based weights from the subset space established, the least squares estimate of $\{x^L_{rt}\}$ is found holding $\{w_{it}\}$ and $\{x^R_{rt}\}$ fixed.

Let $\mathbf{K} : Nn \times p$ be the matrix $k_{rj} = w_{ij} x^R_{sj}$

$$
\mathbf{K}^{(m)} = \begin{bmatrix}
w_{11}^{(m+1)} x_{11}^{R(m)} & w_{12}^{(m+1)} x_{12}^{R(m)} \\
w_{11}^{(m+1)} x_{21}^{R(m)} & w_{12}^{(m+1)} x_{22}^{R(m)} \\
\vdots & \vdots \\
w_{11}^{(m+1)} x_{n1}^{R(m)} & w_{12}^{(m+1)} x_{n2}^{R(m)} \\
w_{21}^{(m+1)} x_{11}^{R(m)} & w_{22}^{(m+1)} x_{12}^{R(m)} \\
\vdots & \vdots \\
w_{21}^{(m+1)} x_{n1}^{R(m)} & w_{22}^{(m+1)} x_{n2}^{R(m)} \\
\vdots & \vdots \\
w_{N1}^{(m+1)} x_{11}^{R(m)} & w_{N2}^{(m+1)} x_{12}^{R(m)} \\
\vdots & \vdots \\
w_{N1}^{(m+1)} x_{n1}^{R(m)} & w_{N2}^{(m+1)} x_{n2}^{R(m)}
\end{bmatrix}
$$

and $\mathbf{L} : n \times Nn$ be the matrix with $l_{ij} = b_{rs,h}$ and $j = n(h-1)+s$

$$
\mathbf{L} = \begin{bmatrix}
b_{11,1} & b_{12,1} & \dots & b_{1n,1} & b_{11,2} & \dots b_{1n,N} \\
b_{21,1} & b_{22,1} & \dots & b_{2n,1} & b_{21,2} & \dots b_{2n,N} \\
\vdots & & & & & \\
b_{n1,1} & b_{n2,1} & \dots & b_{nn,1} & b_{n1,2} & \dots b_{nn,N}
\end{bmatrix}
$$

The least squares estimate for $\mathbf{X}^L$ is found as

$$
\mathbf{X}^{L(m+1)} = \mathbf{L}\mathbf{K}^{(m)}(\mathbf{K}^{(m)T}\mathbf{K}^{(m)})^{-1}
$$

Similarly let $\mathbf{M} : Nn \times p$ be the matrix $m_{rj} = w_{ij} x_{sj}^L$

$$
\mathbf{M}^{(m)} = \begin{bmatrix}
w_{11}^{(m+1)} x_{11}^{L(m+1)} & w_{12}^{(m+1)} x_{12}^{L(m+1)} \\
\vdots & \vdots \\
w_{11}^{(m+1)} x_{n1}^{L(m+1)} & w_{12}^{(m+1)} x_{n2}^{L(m+1)} \\
w_{21}^{(m+1)} x_{11}^{L(m+1)} & w_{12}^{(m+1)} x_{12}^{L(m+1)} \\
\vdots & \vdots \\
w_{N1}^{(m+1)} x_{n1}^{L(m+1)} & w_{N2}^{(m+1)} x_{n2}^{L(m+1)}
\end{bmatrix}
$$

then the least squares estimate for $\mathbf{X}^R$ is given by

$$
\mathbf{X}^{R(m+1)} = \mathbf{L}\mathbf{M}^{(m)}(\mathbf{M}^{(m)T}\mathbf{M}^{(m)})^{-1}
$$

Increase the counter m by one and return to equation (3.31). The process is repeated until convergence of $\mathbf{X}^{L(m)}$ and $\mathbf{X}^{R(m)}$. Since Carroll and Chang point out that convergence is only up to the transformation

$$\mathbf{X}^L = \mathbf{X}^R\mathbf{C}$$

for a diagonal matrix $\mathbf{C}$ with non-zero entries, set $\mathbf{X} = \mathbf{X}^L$ and $\mathbf{G} = \begin{bmatrix} x_{11}^2 & x_{12}^2 \\ x_{11}x_{21} & x_{12}x_{22} \\ \vdots & \vdots \\ x_{11}x_{n1} & x_{12}x_{n2} \\ x_{21}x_{n1} & x_{22}x_{n1} \\ \vdots & \vdots \\ x_{n1}^2 & x_{n2}^2 \end{bmatrix}$
and obtain the final $\mathbf{W} = \mathbf{F}\mathbf{G}(\mathbf{G}^T\mathbf{G})^{-1}$.

The subject space is represented by $\mathbf{X} : n \times p$ and the object space by $\mathbf{W} : N \times p$.

Chapter 4

Technical Components

The practical application of this project is centered around the development of a Graphical User Interface (GUI) designed to perform Multidimensional Scaling. This piece of Software, named the **MDS-GUI**, is the topic of Chapters 5 and 6. The subject of this Chapter is the programming tools used in the development of the **MDS-GUI**. The software was built from scratch using a combination of *tcltk* and *R*, and as such, both of these will be discussed.

4.1 Tcl & tcltk

The *Tcl* language was developed by John K. Ousterhout in 1988 (Ousterhout and Jones, 2010). *Tcl* is a scripting language with primary purposes of development in mind. As a result, it is most commonly used for prototyping, writing scripted applications and Graphical User Interfacing. Due to its open source nature, many developers have contributed to the language and many extensions have been developed for specific applications. One such extension that came about was another creation of Ousterhout, and this was in the form of the *Tk* Toolkit. This *Tk* extension has key applications in the development of user interfaces. Both *Tcl* and *Tk* were developed using the low level *C* language. As a result, the interpreters of the *Tcl* coded script are run through *C* libraries. The term “low level” is used to describe a language that has very little abstraction from the machine language. *Tcl* on the other hand is considered a "high level" language in that there is a high amount of abstraction, i.e. the language deals with virtually no technical machine processes such as handling registers or memory addresses.

4.1.1 Integrating Languages

One of the great advantages of *Tcl* is its ability to integrate with multiple Operating Systems and Programming languages. Code written in a Windows system will therefore be compatible with Linux and Mac. Many of the extension applications written in *Tcl* may be integrated into other coding platforms and used by them. One such example is with the *R* language, which will be discussed throughout Section 4.2. Applications developed in *Tcl* may be 'wrapped' into the *R*-Environment. This means that these *Tcl* libraries may be utilised using *R* code and done so in the *R* syntax. The main reason behind this stems from the fact that both *Tcl* and *R* were developed using *C* and therefore have compatible base code. Full *Tcl* like projects can thus be done in an environment outside of the *Tcl* native environment and in a syntax only somewhat resembling the original. This allows for applications developed in *Tcl* to be able to interact with the functionality of a language such as *R* and make use of the unique capabilities of those languages that was previously unavailable to it. A useful feature of these integrated systems is that while the syntax differs between platforms, the methods and functions of the applications do not. This means that a user who has become proficient in a *Tcl* application in one environment should be able to transition into using it in another environment with relative ease.

4.1.2 tcltk

One language that extends from both *Tcl* and the *Tk* toolkit is called *tcltk* (often written *Tcl/Tk*). The main function of this sub-language is in the development of Graphical User Interfaces with emphasis on rapid development (Ousterhout and Jones, 2010). The *tcltk* language is available not only in the native *Tcl* environments, but has been developed to interface with many other popular languages, such as *Python*, *Perl* and *R*. The emphasis of the Chapter is placed on the *R* wrapped version of *tcltk*, and further information on the appropriate *R* package can be found in Section 4.2.2.3.

4.1.3 Features and Benefits of tcltk

One important characteristic of *tcltk* is in its object oriented programming nature. A brief description of what this means is that the structure in which applications are built in *tcltk* is based on classes and methods. Taking the scenario where the *R* wrapped version of *tcltk* is used; any object created using the *tcltk* package will be of a class *Tcl*. All objects of this class are then compatible with the methods applicable to the *Tcl* class and are able to interact with one another with maximum efficiency. In *R*, a *Tcl* object is created using the `tclVar` function. These objects are unable to interact with non *Tcl* objects directly. In the event that information of an object of that class needs to be accessed for external use, the `tclvalue` function is used. This function extracts the required element from the object in a format interpretable by non *Tcl* objects.

The object oriented nature of *tcltk* allows for efficient and effective programming and development. The application was designed for rapid prototyping and development. This is made possible by the fact that, when using *tcltk*, simple GUIs may be put together quickly and with a minimal amount of code. This is not to say that all products of *tcltk* are developed quickly and are short, it simply means that ideas may be tested with relative ease.

Continuing with the benefits related to interface development, another key advantage to *tcltk* is its ability to test new code and scripts 'on the fly' (Ousterhout and Jones, 2010). This means that it is not necessary to fully compile all scripted code whenever any change is made. So, if a GUI is being developed and is open, when a function relating to a component of the GUI is altered the changes will take effect immediately and should be reflected in the software without having to reload it entirely.

For further discussion on the benefits and features of *Tcl*, *Tk* and *tcltk*, the reader is referred to *Tcl and Tk Toolkit* (Ousterhout and Jones, 2010).

4.2 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 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 mentioned in Section 4.1.2, as well as *R* specific functions were used throughout the practical side of the project. This Section will provide an overview of the *R*-Environment as well as some of the specific libraries used in the development of the MDS software, described in detail in the Chapters to follow.

4.2.1 The *R*-Environment

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. It is for this reason that there will be no elaboration on how *R* works or the syntax of the language. Instead the focus will be on a description of the features and strengths of *R* and on what the language is based. The developers of *R* and the RGui describe the product as "a fully planned and coherent system" and hence believe the term 'environment' to be appropriate. Furthermore they describe the suite of software facilities for data manipulation with a few key summarising features. According to the R-project homepage (R Development Core Team, 2012), these features are:

- An effective data handling and storage facility.
- A suite of operators for calculations on arrays, in particular matrices.
- A large coherent, integrated collection of intermediate tools for data analysis.

- Graphical facilities for data analysis and display either on-screen or on hard copy.
- A well-developed, simple and effective programming language which includes conditionals, loops, user-defined recursive functions and input and output facilities.

The open source feature is another that deserves mention. 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. More on packages in Section 4.2.2.

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.

Most coding done using *R* makes use of what are known as "S3" objects, which have no formal definitions and are able to be used somewhat freely between other S3 objects. More advanced users however are able to make use of "S4" type objects, which are more formally defined. The use of S4 objects complies with the class based "object orientated programming" (OOP) methods such as those utilised when using a language such as *Java*. S4 objects are built specifically for use of methods of their class and are only able to interact with others objects of the same class.

4.2.2 *R*-Packages

A number of external packages were put to use in conjunction with unique code in the practical processes of the project. This Section will provide an overview of the main packages that were used. It is not uncommon for packages designed for *R* to have literally dozens of functions, which is why only the functions most used and relevant to the practical applications of this project will be discussed. In all cases where the package is manually downloaded and installed, the version used during development will also be provided.

The **base** package (R Development Core Team, 2012) is used regularly by all *R* users and contains the bare essential operations common to any coding language.

4.2.2.1 stats package

The **stats** package is one of the basic packages of *R*, meaning that it comes as a default and does not require external download. The package was written and maintained by the R Development Core Team. While the package comes pre-installed, the library does need to be added prior to making use of any of its functions. The package is vast in the number of functions that it contains and is host to the functions that perform a large portion of the basic statistical operations. As expected, some of the basic statistical functions such as `cov`, for computing the covariance matrix, are used regularly throughout the code. These basic functions are however self explanatory and used regularly enough in practice that in-depth explanations are not required.

A function from the **stats** package that does require elaboration, however, is the `cmdscale` function. This function performs the basic Classical Scaling (or Principal Coordinate Analysis) on a symmetric matrix of dissimilarity data. The user is able to control, p, the number of dimensions in which to perform the MDS. The resulting values produced by the function is a set of coordinates in the p dimensional Euclidean space. These coordinates may be graphically displayed using a simple plotting function, such as `plot` from the *R* **base** package. The complete listing of the resulting eigen values are also available, if specified by the user.

4.2.2.2 MASS package

The **MASS** package, similar to the stats package, is another of the basic default packages of *R*. **MASS**, maintained by Venables and Ripley (2002), stands for "Modern Applied Statistics with S". The package holds over 80 data sets and the functions within the package are more specialised than those of the **stats** package.

Four functions from the **MASS** package were put to great use in the MDS software. The first two of these perform MDS operations and are the `isoMDS` and `sammon` functions. The `isoMDS` function is used for performing Kruskal's Non-metric Multidimensional Scaling while the `sammon` function is used for the Sammon mapping procedure. Both of these functions are utilised in a similar way. In both cases the user is required to provide a symmetric matrix of dissimilarities or relative distances and p, the number of dimensions in which to perform the MDS. Additional options are also available to the user which relate to the iterative nature of these two forms of MDS. Firstly, a starting configuration of the users preference may be provided to the function if the default form, the result of Classical Scaling, is not satisfactory. There is also an option for the maximum number of iterations and the tolerance to be adjusted. The `sammon` function, in addition, does provide adjustability to the "magic" component which is explained in Section 3.3.2. The output of the two functions are similar to the `cmdscale` function discussed above in that they produce p vectors of coordinates. In addition to these coordinates, a stress value of the final configuration is also provided. Since a new function for calculating stress in a number of formats was written specifically for this project, the stress capabilities of the functions were not applicable.

The third **MASS** function that was used extensively was the `Shepard` function. This creates the coordinates relating to the Shepard Plot discussed in Section 2.6.2 . The input is simply the proximity matrix used for the MDS procedure and the resulting configuration in coordinate form.

Finally the `isoreg` function is put to use in performing the isotonic regressions appropriate to Non-Metric MDS methods. The input of the function is the matrix of distances between configuration points $\mathbf{\Delta}$. The output is then the corresponding transformed $\hat{d}$ used in plotting the transformation curve onto the Shepard plot and in calculating stress for Non-Metric MDS procedures.

4.2.2.3 tcltk package

The *R* integrated version of *tcltk*, discussed in Sections 4.1.2 and 4.1.3, is housed in the *R* package named **tcltk**. The **tcltk** tool pack comes standard with any *R* version and is maintained by the *R* Development Core Team. The tool pack is treated slightly differently to most of the packages used by *R*, in that it is an entirely integrated version of tools developed for another language. A minor result of this is that the tool pack is loaded using the `require` command whereas most other packages are called using the `library` command. Another consequence of the difference is that there is virtually zero documentation freely available for the *R* wrapped commands of **tcltk**.

Literally dozens of functions calling upon the applications and widgets of the *tk* GUI toolkit are utilised extensively throughout the thousands of lines of software base code. These functions will not be elaborated upon here, but each will be identified and discussed at length in the Chapters of this dissertation that describe the software that has been developed, namely Chapters 5 and 6. This will be done in such a way that when features of the software are mentioned, the relevant functions behind them will be referenced and explained.

Within the **tcltk** package are further sub-packages that may be accessed by *R* which provide additional widgets and functions. Once **tcltk** has been required into the *R*-Environment, these sub-packages need to be further loaded into the environment. The following sub-packages, discussed below, are loaded to *R* using the `tclRequire` function. As the name of the function suggests, the requiring function is itself a function of the **tcltk** package. These packages are kept as optional additions due to the fact that each provides specialised tools which may not be required by a large portion of *tcltk* users.

- **Tktable**

  The **Tktable** extension of **tcltk** provides useful additions to the visual table widgets that come standard with **tcltk**. This add-on is appropriate to any user wishing to create tables with increased variability in the layout, or who wishes to make use of more advanced tools, such as including scroll bars to a table. As the software of this project makes extensive use of tables and requires them to have numerous formats, the **Tktable** add-on is essential.

- **BWidget**

  The next add-on package, being the **BWidget** package, gives the *tcltk* user access to further general tools of many different applications. The expansion is also essential to a user who is creating a GUI with multiple levels, i.e. has software that allows additional toplevel windows to be opened from the main toplevel window. Without the **BWidget** expansion, opening secondary toplevel windows tends to result in errors. More on toplevel windows in Chapter 5.

The following three packages, being **tkrplot**, **rpanel** and **tcltk2** are all individual packages that are linked to the use of tcltk in *R*. While these three packages are similar to **Tktable** and **BWidget** in that they are extensions to the basic **tcltk** capabilities, they are different in that they are stand alone packages that are not included in the default *R* library.

4.2.2.4 tkrplot package

The **tkrplot** package is essential to any tcltk user developing user interfaces that incorporates *R* plots as graphics. Any software dealing with Multidimensional Scaling will aim to output a graphical representation of the ordination result, which is why the **tkrplot** package is required for this application. The package **tkrplot** was written and is maintained by Tierney (2011). Development of the MDS-GUI utilised **tkrplot_0.0-23**.

The package makes use of two primary functions. These two functions are `tkrplot` and `tkrreplot`. The first function, `tkrplot`, is called when a *tcl* plot object is to be created. When a new *tcl* plot object is defined, it usually entails naming the object and providing the method describing the plot with a relevant *R* plotting function. Vertical and horizontal scaling options are also available to the user. This plot object may be displayed almost anywhere in the GUI with the use of the `tkplace` function. The second function, `tkrreplot`, was created as a means of updating *tcl* plot objects that have already been created using the `tkrplot` function. For instance, consider a plot object being created with the parameters being to plot object X versus object Y. If at some point the values of X and/or Y change, using the `tkrreplot` function will plot the new values on the same plot object under the same configuration under which it was created. This feature is particularly useful when graphically portraying the results of an iterative procedure, such as MDS.

4.2.2.5 rpanel package

Rpanel provides a set of functions to build simple GUI controls for R functions. These are built on the tcltk package. Uses could include changing a parameter on a graph by animating it with a slider or a "doublebutton", up to more sophisticated control panels. Development of the MDS-GUI utilised **rpanel_1.0-6**.

4.2.2.6 tcltk2 package

The **tcltk2** package, written and maintained by Grosjean (2011), is an important add-on for the **tcltk** package and should also been seen as an upgrade to it. While **tcltk2** does have new features and offer widgets not included in the standard package, it also provides upgraded versions on a number of the existing widgets. Similar to the functions and widgets associated with the **tcltk** package, these will be elaborated on as needed throughout Chapter 5. Development of the MDS-GUI utilised **tcltk2_1.1-5**.

4.2.2.7 RColorBrewer package

Not all packages used in the practical application had intense computational purposes. The **RColorBrewer** package was put to use from a visual perspective. The package was authored and is maintained by Erich Neuwirth and specialises in providing palettes of colours for graphical purposes with emphasis on thematic maps (Neuwirth, 2011). These palettes are based on the palettes compiled by Brewer and Harrower (2002) which were designed with various applications in mind. There are three types of palettes in the package, sequential, diverging and qualitative. Sequential palettes are appropriate when the visualisation is required to indicate the level of an attribute. This is usually done with light colours indicating low values and darker colours indicating higher values. Diverging palettes will give visual indication of objects that are at either extreme of a spectrum. That is, mid range values will be light, where low and high values will both be darker. Finally, Qualitative palletes have no sequential relationship between the gradient of the colours and are thus appropriate for categorical data. These palletes are specifically designed to hold colours where each is as different as possible to any other in the pallete so as to be distinguishable.

Three specific palettes were utilised, namely `Set1`, `Paired` and `Dark2`, all three of which are Qualitative by design. `Set1` and `Paired`, comprising 9 and 12 colours respectively, were combined to form a larger palette for representing objects with categories in MDS configurations. The `Dark2` palette, having eight colours, is used to distinguish between the underlying variable axes shown on the MDS configuration. Development of the MDS-GUI utilised **RColorBrewer_1.0-5**.

4.2.2.8 boot package

The **boot** package offers a host of data sets and functions related to the bootstrapping procedure. Boot was written and is maintained by Canty and Ripley (2010) and the content of the package is based almost entirely on the book "Bootstrap Methods and Their Applications" by (Davison and Hinkley, 1997). Development of the MDS-GUI utilised **boot_1.3-4**.

While no bootstrapping related functions were relevant to this application; the package does contain a useful function for calculating the weighted correlation coefficient between two sets of data, `corr`. This was found to be a convenient and appropriate way of calculating the correlation coefficient between the input proximity matrix and the ordination distance matrix. This correlation figure is then used as an alternative to the stress value in terms of determining the goodness-of-fit of a configuration. This statistic is of course interpreted in an inverse manner to that of stress, in that higher values indicate better fit, with a maximum value of one.

4.2.2.9 RGL package

OpenGL is an open source group of softwares that specialise in 2D and 3D renditions of plots. The product is available as a stand alone program and has also been incorporated and integrated into other environments. **RGL** is the *R* based interface of the widely used *OpenGL* (SGI, 2012) and acts as a 3D engine for the *R*-environment. The *R* version was written by Adler and Murdoch (2011) and is maintained by Murdoch. Development of the MDS-GUI utilised **rgl_0.92-798**.

The capabilities of **rgl** within *R* is vast and itself is the subject of multiple research papers. As a result, the use of the 3D visualisation capabilities in the scope of this application was minute. In the event that the MDS configuration is produced with p equal to three, an option to output the configuration in an interactive 3D format is given. In this case, a simple 3D scatterplot is created by the **rgl** package. This is done using the `plot3d` function. Furthermore, if this is to be done using the text labels as points on the graph, the `text3d` function is used.

4.2.2.10 scatterplot3d package

scatterplot3d is a visualisation package that creates a 3D rendition of scatterplots on a 2D plane. The **scatterplot3d** package was written and is maintained by Ligges and Mächler (2003). The three dimensional effect on a two dimensional plot is achieved with the use of a grid like plane and adjustable angles between axes. There is also the option to highlight points depending on their depth in the third dimension. Therefore, objects closer on the 3rd dimension will be lighter and those further away will be darker. The sole function of the package is itself called `scatterplot3d`, however the function has a possible 41 adjustable parameters which may be tweaked until the visualisation most appropriate to the user is achieved. Development of the MDS-GUI utilised **scatterplot3d_0.3-31**.

4.2.3 Other R-Packages of Interest

All packages mentioned to this point were used throughout all practical software development. There are, however, a few other packages that have specific MDS related functions and as a result should be mentioned in this Section. While the functions of the following packages were not involved in the final format of the code for this project; each was experimented with and the output results were compared closely to the corresponding functions that were finally included.

4.2.3.1 smacof package

The **smacof** package provides a comprehensive range of Multidimensional Scaling methods based on stress minimisation by means of majorisation. SMACOF algorithms are discussed in detail in Chapter 3. The package, written and maintained by de Leeuw and Mair (2009b), provides a number of the different smacof approaches. These include, (Mair and de Leeuw, 2008): Simple smacof on symmetric dissimilarity matrices; smacof for rectangular matrices (unfolding); smacof with constraints on the configuration; three-way smacof for individual differences; and spherical smacof. Each of the forms are available to be performed in both a metric and a non-metric manner. The package also places great emphasis on graphical portrayal of the MDS results and incorporates features of both the **rgl** and **scatterplot3d** packages already discussed in Section 4.2.2.9 and 4.2.2.10 respectively.

While the **smacof** package provided great use during the development stages on the MDS application of this project, it was not utilised in the final version of the software. The code performing the SMACOF functions in the **smacof** package calls upon functions written using the *C* language, and therefore adaptation of this code (as required by use of the MDS-GUI) would not have been straightforward. The SMACOF function was rewritten in *R* for both the metric and non-metric algorithms with the help of le Roux (2012). The use of original code allowed for the easy incorporation of *tcltk* functionality.

4.2.3.2 homals package

The **homals** package (de Leeuw and Mair, 2009a) contains functions for performing homogeneity analysis. The primary function of the package is `homals` which performs homogeneity analysis and returns an S4 object of the "homals" class. Useful plotting functions geared to using "homals" objects include `plot.homals` and `predict.homals`. Future versions of the MDS-GUI intend to make use of an adaptation of the code from this function for performing the Gifi method of MDS. The necessity to adapt the code rather than use the function itself comes from the importance of incorporating *tcltk* components to the methods, which is made impossible by the class system in place in the function. Specifically, the functions to be adapted for use by the MDS-GUI will include: `homals`, `checkPars`, `orthogonalPolynomials`, `weigthedGramSchmidt`, `normX`, `centerX`, `computeY`, `totalLoss`, `sumSet`, `updateY`, `restrictY`, `nominalY`, `totalSum` and `expandFrame`.

4.2.3.3 SensoMineR package

SensoMineR (Husson et al., 2011) is a package that has assembled a host of methods for analysing sensory data. The package was developed such that it aims to output graphical results that are easy to interpret. Among the many functions and tools found in the package, the function of interest to this application is the `indscal` function which performs the INDSCAL method of MDS with specific design for use on the accompanying "napping" data.

Future versions of the MDS-GUI will incorporate INDSCAL. The code for which will also be developed with the help of le Roux (2012).

4.2.3.4 Limma package

An increasingly popular application of multivariate analysis and ordination techniques is in the study of microarray data. Many *R* packages have been specifically designed for the analysis of microarray data, and the **Limma** package, by Smyth (2005), is one of these. Within the package is the `plotMDS` function which has been specifically designed to perform Classical Scaling on microarray data and automatically plot the results in an appropriate format.

4.2.3.5 Labdsv package

Another field that has come to utilise the benefits of ordination methods is that of ecology. The **Labdsv** package, written and maintained by Roberts (2010), is a package designed for the statistical analysis of ecology data and contains numerous ordination methods for doing so as well as means of graphically portraying the results. The package itself is written in the S4 class system of *R*, which is a somewhat more advanced way of writing *R* code. As a result, all objects created by functions within the package need to be of the same class in order to interact in an efficient manner. The `nmds` function of the labdsv package is a wrapped version of the `isoMDS` function of the **MASS** package, which was discussed in Section 4.2.2.2. The term "wrapping" in this case refers to how the function has been adapted to the S4 class system of the package and will allow the output of the function to be objects of an appropriate class. Results of the `nmds` function are thus able to interact with other objects created by functions in the package in a manner that the programmer intended.

4.2.3.6 vegan package

The **vegan** package is another that has been developed for *R* with the purpose of providing analysis techniques for ecology data. The package is maintained by Oksanen et al. (2011). Vegan contains a function called `metaMDS` which conglomerates a number of other functions to produce a reliable MDS result. The core MDS calculation is once again done using the `isoMDS` function of the **MASS** package. `metaMDS` however tries to find the most stable solution by using several random starting configurations for the procedure, using `initMDS`, and monitoring the resulting stress values. The output is then scaled for convenience of interpretation and, what the developers refer to as, "species scores" are then added to the configuration using the `postMDS` function.

Chapter 5

The MDS-GUI

The practical application of this Masters project was the development of an *R* based piece of software, named the 'MDS-GUI'. The name is an acronym for 'Multidimensional Scaling Graphical User Interface'. This Chapter will describe the details of the MDS-GUI in terms of the design and features of the software. Chapter 6 will then focus on its application capabilities. The GUI will be available for download from the 'CRAN' website for *R* in the package called **MDSGUI**. All supporting documentation relating to this *R* package can be found in the Appendices.

5.1 An Introduction to the MDS-GUI

The *R* language is growing in popularity, not only among statisticians, but throughout many fields of research. This is due to the broad spectrum of data analysis which has potential applications in many scientific and financial sectors of research. Within this growing popularity, the use and development of GUIs is fast becoming a popular means of statistical processing in the *R* community. This is due to the fact that many researchers benefit from a simple interface with point and click capabilities. The MDS-GUI was thus designed to provide such assistance to researchers wanting to perform Multidimensional Scaling procedures. This piece of software allows the user, with no theoretical background on the subject, to perform a number of MDS procedures and output a host of relevant details and graphics.

These graphical outputs are of an interactive nature, allowing users to make adjustments with the use of a computer's cursor.

In broad terms the MDS-GUI allows the user to simply and efficiently input their desired data, choose the type of MDS they would like to perform as well as select the type of output they would like to achieve by the analysis. The use of sub-menus and property tabs give the user the option to fine tune the specific parameters of the desired MDS procedure as well as provide options to alter the way in which the resulting plots are displayed.

The development of the GUI itself was done using the *R* wrapped *tcltk*, details of which can be found in Chapter 4. Similar software has been developed using *tcltk* in *R* for other multivariate processes such as the BiplotGUI (la Grange et al., 2009) and the caGUI (Markos, 2010) to name two. These previously developed softwares have allowed for easy implementation of complicated concepts so as to allow a user to obtain thorough results quickly. The MDS-GUI was developed with the same intention, that is for the menu structures and overall layout be set out in a way that has been found to be user friendly and uncomplicated as well as comprehensive and effective.

5.1.1 A Tour

The MDS-GUI has a layout with various sections. Figure 5.1 below shows the default view of the GUI with each of its major sections labeled one to five. A description of each of these areas will now be given in general terms. Where appropriate, when features require elaboration, details will be provided in Section 5.4.

1. **Main Plotting Area**: The area labeled as 'one' is the area on which the MDS based configuration is shown when $p = 2$, which is the default. In other words, the Euclidean distances, d_{rs}, are illustrated in area one. The aspect ratio of the plotting area is one thus preserving the interpretation of the distances regardless of the orientation of the axes.

2. **Plotting Tabs**: Label two refers to the tabs on top of the main plotting area. The user has the option to make use of up to five plotting

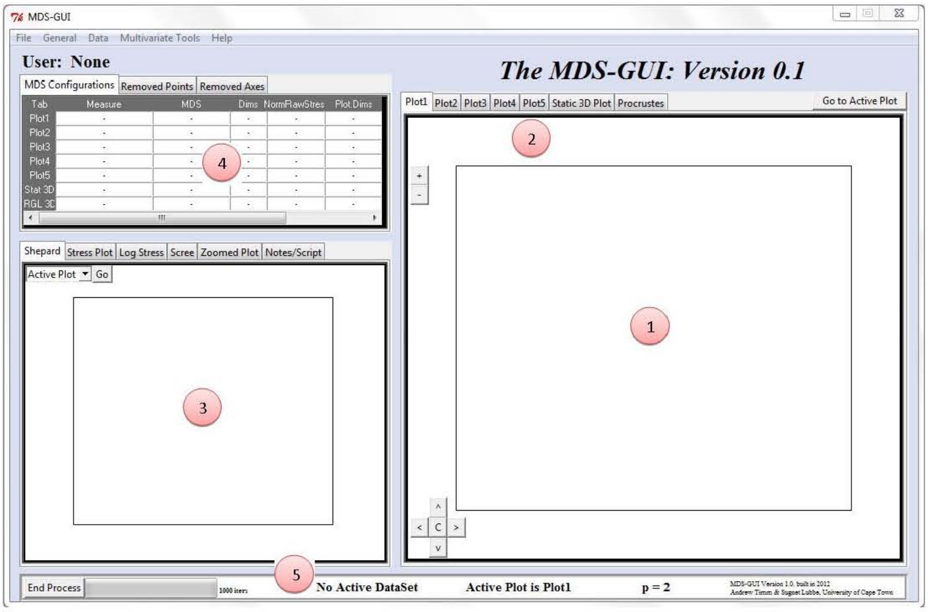


Figure 5.1: The MDS-GUI

tabs to perform independent MDS procedures with individual settings. These results may then be compared. In addition, the 'Static 3D Plot' tab shows the output of the two dimensional depiction of the result when $p = 3$ and the 'Procrustes' tab shows the result of a Procrustes analysis between two separate configurations.

3. **Secondary Plotting Area**: The 'Secondary Plotting Area' of the MDS-GUI is labeled 'three'. This smaller area houses multiple diagnostic outputs generated as a default for most of the MDS processes. Each of the Tabs of the area show a different utility. These tabs are labeled as: 'Shepard', 'Stress Plot', 'Log Stress', 'Scree', 'Zoomed Plot' and 'Notes/Script'. The functionality of each of these will be discussed in full in Section 5.4.

4. **Table Section**: The window on the top left of the GUI holds the relevant tables included in the software. The front most table is called the 'MDS Configurations' table and holds important information relating to each of the MDS configurations in all main plotting tabs. The second and third tables are called the 'Removed Points' and 'Removed

Axes' tables respectively. Features of the MDS-GUI include the option to remove points from the configuration and also remove variable axes from the display. The information contained in these tables pertain to these scenarios. See also Section 5.4.13.

5. **Information Pane**: The final numbered label refers to the Information Pane located at the bottom of the GUI. This area displays various information relevant to the applications of the user. The pane includes information regarding the data set being used, the current plotting area and the software developer details.

5.2 Multidimensional Scaling Capabilities

The MDS-GUI provides six different methods of Multidimensional Scaling to the user. Refer to Chapter 3 for mathematical details regarding each of these methods. 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 $n \times n$ dissimilarity matrix $\mathbf{\Delta}$ as the input. The MDS-GUI however handles all necessary input management automatically, and the user may simply choose their desired method regardless of the form of data that is uploaded.

5.3 Menu Structures of the Software

The MDS-GUI was designed with a number of menu structures included. The various menus throughout the software control a host of aspects that effect the output and presentation of the applications.

5.3.1 Top Menu

The Top Menu of the MDS-GUI is in a format common to many modern day windows based softwares. The menu system is accessible via the topmost pane of the GUI and consists of five drop down menu lists, each of which will be shown and the features described.

5.3.1.1 File Menu

The *File* menu is shown in Figure 5.2. Like most 'File' type menus, the menu focuses on the handling of the specific files and workspaces of the user. Actions available to the user from this location include: setting up a user profile; saving and loading workspaces; printing the focused plot; clearing all current data and output; and finally exiting the program in a safe manner. Exiting the program also gives the option to save the workspace.

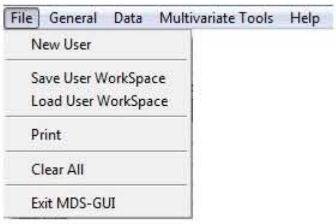
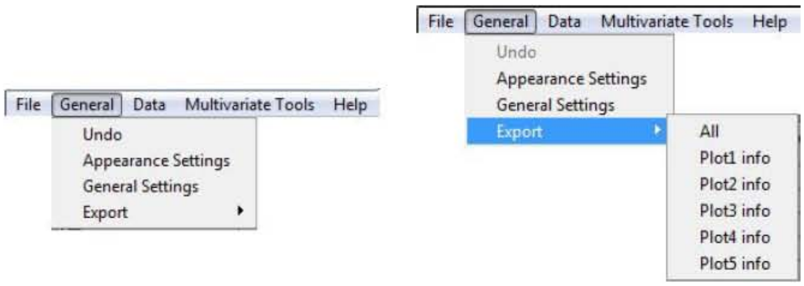


Figure 5.2: The File Top-Menu

5.3.1.2 General Menu

The *General* menu, in Figure 5.3(a), contains a broad range of options. The first item on the menu is *Undo*, which reverses the most recent action performed by the user. The capabilities of *Undo* are currently limited to the most recent point moving alteration made to the active configuration. The *Undo* option will only be available when a recent point alteration has been made and will revert the configuration back to the state prior to the alteration. When not available, the option will be greyed out. The *Appearance Settings* item opens the window, shown by Figure 5.4, which gives the user the option to adjust the background colour of both the main GUI and all pop-up windows. Studies have suggested that the appeal of colour used by computer software has profound effects on the user (Martin, 1993), and this option allows the user of the MDS-GUI to make adjustments to suit their personal liking.

The *General Settings* item is used to access the General Settings pop-out menu, which is discussed in full in Section 5.3.4. Finally the *Export* submenu, which is displayed in Figure 5.3(b) gives a list of the various export options available to the user.



(a) General Menu (b) Export Submenu

Figure 5.3: The General Top-Menus

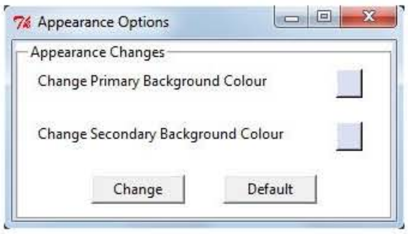
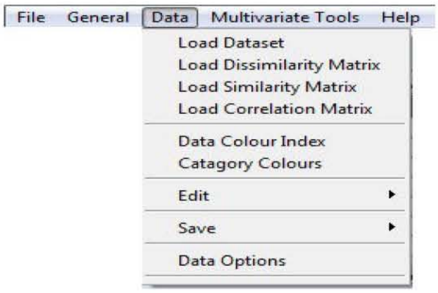


Figure 5.4: Appearance Settings

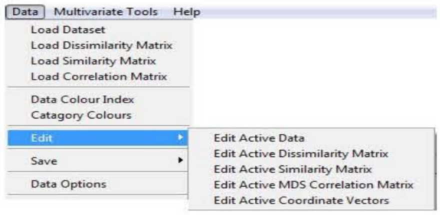
5.3.1.3 Data Menu

The *Data* menu provides all options relating to the user upload of data on which the MDS procedures are performed. Chapter 2 described how the data may originally be provided in a number of different forms, all of which are allowable by the MDS-GUI. The user is able to upload: a samples by variables matrix $\mathbf{Z}$:$n \times m$; a distance matrix $\boldsymbol{\Delta}$:$n \times n$; or an $n \times n$ similarity or correlation matrix. In the case where $\boldsymbol{\Delta}$ is not uploaded, appropriate calculations and transformations are performed automatically in order to construct $\boldsymbol{\Delta}$, which is used in all MDS procedures. Details on the settings of these calculations are provided in Section 5.3.1.4.

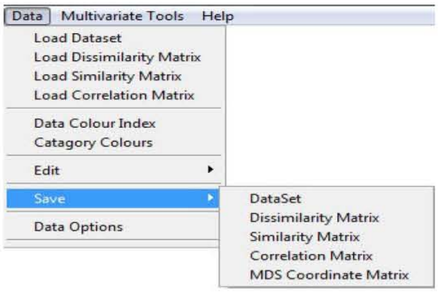
Uploading into the MDS-GUI makes use of the standard windows explorer Load/Save window that users are familiar with. Upon choosing the file in which the data is stored, the user is presented with the *New Active Dataset Options* window (Figure 5.6), which contains various options available regarding the data. Firstly, the user is prompted to name their data. This name will not only be displayed in the *Information Panel* but will also



(a) Data Menu



(b) Edit Submenu



(c) Save Submenu

Figure 5.5: The Data Top-Menus

be presented in the title of all configuration output. Simply pressing the 'Enter' key bypasses any naming if unwanted by the user. The second option gives the option to transpose the data (This only applies when $\mathbf{Z}:n \times m$ is uploaded). All procedures throughout the package require that objects be rows and variables be columns, therefore if the data comes as the transpose of this it may be corrected. Following this, the user may scale their data such that all values range between zero and one. Finally, an allowance is made for the data to have a categorical column. When such a column exists, the user may choose its location. This column in the data is then removed from the rest and treated independently whereby the various categories are identified and colour coded. Further details on data categories are provided in Section 5.4.11. Whenever any data is uploaded to the MDS-GUI a series of standard checks takes place. All files are checked to be of an appropriate format and data is then checked whether it contains only numeric values that are non 'NA'. In the cases of uploading a (dis)similarity matrix, the matrix is tested whether the number of rows equals the number of columns. If any check fails, an appropriate error message is displayed and the user is prompted to reattempt the upload. If any elements of an uploaded **Z** matrix are less than or equal to zero, some metric options are made unavailable for calculation of the dissimilarity matrix (See Section 5.3.1.4).

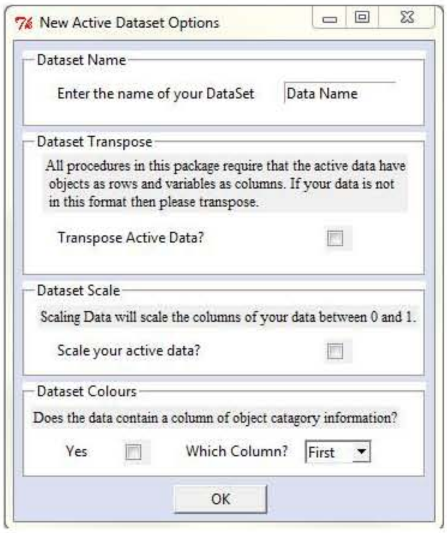
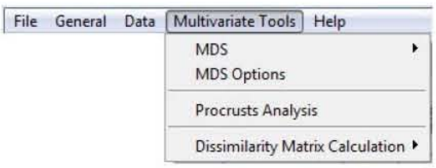


Figure 5.6: Uploaded Data Menu

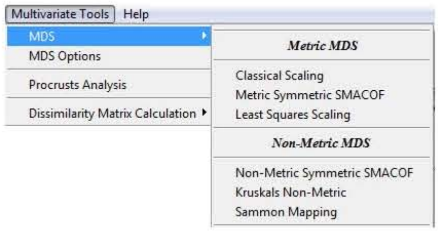
The next two items, *Data Colour Index* and *Category Colours*, relate to the colours assigned to each object of the data. Details on the use of these options is provided in Section 5.4. The *Edit* option (Figure 5.5(b)) of the *Data* menu allows the user to make manual adjustments to all active data within the GUI, including the $n \times p$, $\mathbf{X}$ matrix. Any adjustments made are immediately represented on the MDS configuration in the main plotting area. *Save* (Figure 5.5(c)), similar to the *Load* options, makes use of the Windows Load/Save window to save any of the active data to an external file in any desired format. The final item of the menu is *Data Options* which recalls the Data Options Menu described in Section 5.3.5.

5.3.1.4 Multivariate Tools Menu

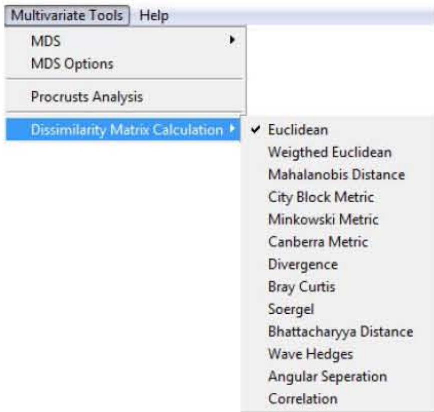
The menu that provides access to the multivariate functionality of the MDS-GUI is entitled *Multivariate Tools*. The *MDS* submenu is shown in Figure 5.7(b) which lists the various forms of MDS that are available in the GUI. The list is segmented into four groups, being Metric MDS, Non-Metric MDS, alternative MDS models. Refer to Chapter 3 and Section 5.2 for a discussion on these forms of MDS. The list items are only available when valid data has been uploaded into the software (an error message is returned when no valid data is present). Upon selection, the chosen MDS procedure is performed on the active $\mathbf{\Delta}$ matrix. Following this, the item *MDS Options* recalls the MDS Options menu which is the subject of Section 5.3.6.



(a) Multivariate Tools Menu



(b) MDS Submenu



(c) Dissimilarity Calculation Submenu

Figure 5.7: The Multivariate Tools Top-Menus

The next available multivariate tool is Procrustes Analysis and is called from the *Procrustes Analysis* menu item. This feature is described in full in Section 5.4.6. Finally, the *Dissimilarity Matrix Calculation* submenu is shown in Figure 5.7(c). This menu displays the list of all possible methods for calculating $\mathbf{\Delta}$ when the uploaded data is in the form of $\mathbf{Z}$:$n \times m$. The list makes use of the 'radiobutton' tool of *tcltk* where only one item of the list may be selected at a time, with the active item identified with a tick mark. The default measure is the Euclidean Metric. If the measure is changed, $\boldsymbol{\Delta}$ is recalculated and then used for all subsequent MDS procedures. Any MDS mapping performed before this change is unaffected as this allows the user to make direct comparisons to the results under different distance metric calculations. Upon calculation of $\boldsymbol{\Delta}$, certain checks are performed on the result. Some MDS methods require specific attributes of the input data, for example Kruskal's Analysis and Sammon Mapping may not have negative elements in the matrix. When criteria are not met, the appropriate MDS methods are grayed out and made unavailable for the current $\boldsymbol{\Delta}$.

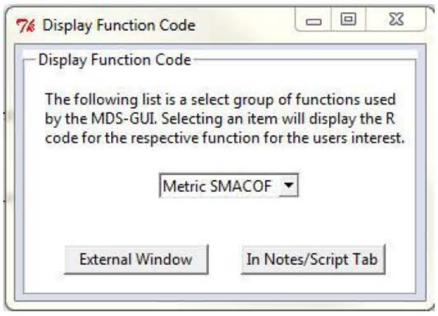
5.3.1.5 Help Menu

The *Help* menu, shown in Figure 5.8, provides the user with assistance and details regarding the MDS-GUI itself. The first item on the menu, *Function Code* recalls a menu which lists a comprehensive assortment of the most important functions used within the MDS-GUI. Figure 5.9(a) shows this

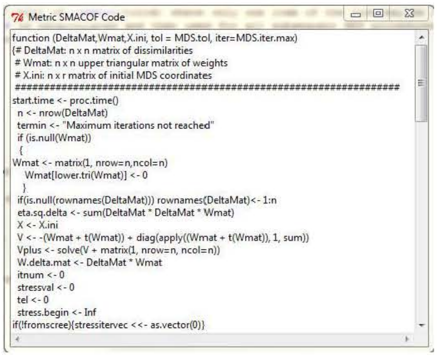
Figure 5.8: The Help Top-Menu

menu where the drop down box (*tcltk* ComboBox) currently has 'Metric SMACOF' selected. The user is given the option to display the code for this selection in either a Pop-Out text box, or in the built in *Notes/Script* tab located in the *Secondary Plotting Area*. The resulting output of these for the Metric Smacof function code are displayed in Figures 5.9(b) and 5.9(c) respectively. It should be noted that in both cases, the code is displayed for the users interest only and alterations will not effect the function in any way.

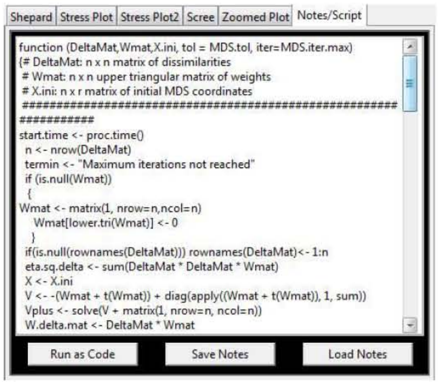
The *Display Pop-Out Help* item is a *tcltk* 'checkbutton', with a tick mark indicating the option has been selected. When the option is activated,



(a) Function Help Menu



(b) Function Help: External



(c) Function Help: Tab

Figure 5.9: Function Help

hovering the mouse cursor over certain areas of the MDS-GUI will cause a window of text to appear with instructions relating to the area under focus. Areas with pop-out help include each of the main plotting tabs, each of the secondary plotting tabs and each table in the table area. Following this, the *Vignette* and *User Manual* options direct the user's default web browser to the **MDS-GUI** package vignette and user manual pages respectively in the CRAN website. Similarly, the *Homepage* item directs the browser to the MDS-GUI R-forge home page (*Note, websites will only be activated when **MDS-GUI** is successfully available on CRAN*). Finally, the *About* option opens a text box (Figure 5.10) with information regarding the software. The information includes: the version of the MDS-GUI, developer information, developer contact details and license information (*Note, License information will also only be available post CRAN submission*).

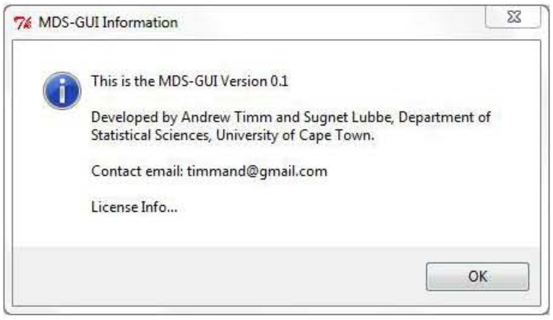


Figure 5.10: About: Text Box

5.3.2 Main Plot Menu

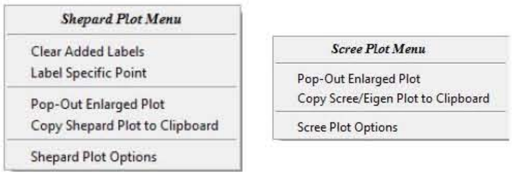
Figure 5.11 shows the menu common to each of the five configuration plotting areas. The menu in focus corresponds to the first plotting tab as indicated by the heading. In each case, the menu is called with the use of the right click button of the mouse when the user's cursor is over the plotting area.

Figure 5.11: Main Plot Options Menu

These menus act as the primary source of actions available to the user, and as such, the details of the functions are discussed in Section 5.4. In general however, the menu provides the user access to features involving: labeling of configuration points, manual manipulation of configuration points, point colour options and variable axes options. The final element of the menu is the *Plot Options* entry which directs the user to the menu described in Section 5.3.7.

5.3.3 Secondary Plot Menus

Each of the plots housed in the *Secondary Plotting Area* has an individual menu that, similar to the *Main Plot Menu*, is accessed by right clicking the plot itself. Figure 5.12 gives the menus for the Shepard Plot and Scree Plot as examples. The menu pertaining to the remaining tabs are similar. Each menu provides access to the plot's specific visual options menu, described in Section 5.3.7, and gives option to pop-out and copy to clipboard. The Shepard Plot menu also provides labeling options described in Section 5.4.3.1.



(a) Shepard Plot Menu (b) Scree Plot Menu

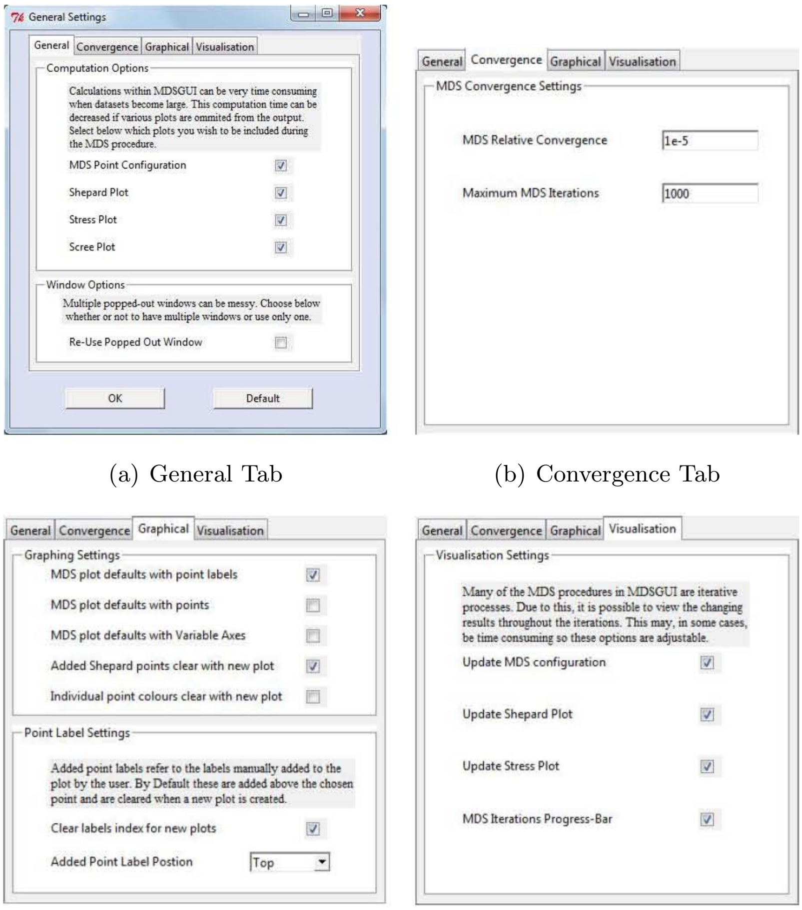
Figure 5.12: Secondary Plot Options Menu

5.3.4 General Settings Menu

The *General Settings* menu, accessed through the *General-Top* menu, controls various technical aspects of the MDS-GUI. The four tabs of the menu are shown in (a) to (d) of Figure 5.13.

5.3.4.1 General Tab (a)

The first tab of the menu contains *Computation Options* and *Windows Options*. By default, the MDS-GUI will calculate all four major processes automatically, these being: the MDS configuration, the Shepard Plot, the Scree Plot, and (when applicable) the Stress Plot. In *Computation Options* the option is available to deactivate any of these computations for all subsequent use. This option is expected to be exercised when data is sizable and all computations have proven to be excessively time consuming.



(a) General Tab (b) Convergence Tab

(c) Graphical Tab (d) Visualisation Tab

Figure 5.13: General Settings Menu

The *Windows Options* pertains to the various popped out windows that might be observed through use of the GUI. By default, any number of these popped out windows may exist. The alternative setting, adjustable here, is that any popped out window replaces the last, thus only one will exist at any time.

5.3.4.2 Convergence Tab (b)

The *Convergence* tab contains options relating to the allocation of computational resources dedicated to the MDS procedures. The *MDS Relative Convergence* refers to the tolerance abided to by the procedures. Secondly, *Maximum MDS Iterations* allows adjustment to the maximum iterations permissible to processes. It should be noticed that the maximum iterations figure is shown on the *Information Panel* adjacent to the *Progress Bar*. See Chapter 2 for details and effects of differing tolerance levels and iteration capping.

5.3.4.3 Graphical Tab (c)

The third tab of the menu, *Graphical*, relates to default options of the visual MDS outputs. *Graphical Settings* is used in conjunction with *Plot Options Menus* in Section 5.3.7, where details of the settings are found. *Point Label Settings* controls point labels manually added by the user. The options include the option to clear added labels or make them follow through to each plot, and allocate their position around the point itself. Further details on point labeling are found in Section 5.4.7.

5.3.4.4 Visualisaiton Tab (d)

The *Visualisation* tab specifies the elements of the MDS procedures that should have its iterative nature depicted visually. This should not be confused with the *Computation Options* elements in the *General* tab. The outputs that have been unchecked will still be processed (provided they have not been deactivated), but will not have their respective plots updated after each iteration. In this case, the final result is plotted upon process completion.

5.3.5 Data Options Menu

The *Data Options Menu* (Figure 5.14) is accessed via the *Data TopMenu*. The menu is similar to the *New Dataset Options Menu* in Figure 5.6, with the difference that this menu may be called at any point and have the settings changed. The displayed tab is associated with an uploaded $\mathbf{Z}$:$n \times m$ matrix or $\mathbf{\Delta} : n \times n$ matrix.

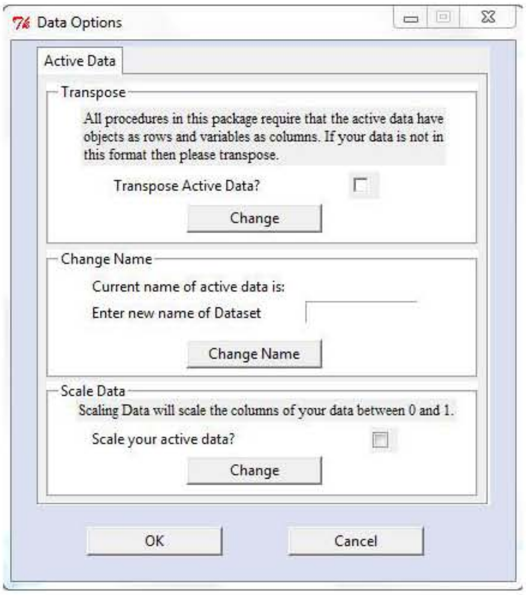


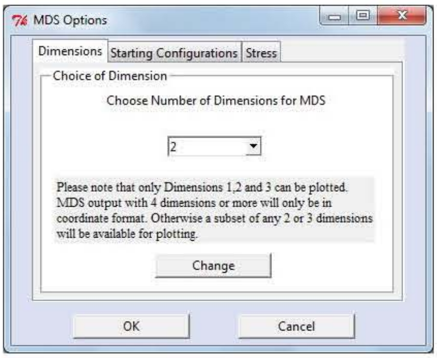
Figure 5.14: Data Options Menu

5.3.6 MDS Options Menu

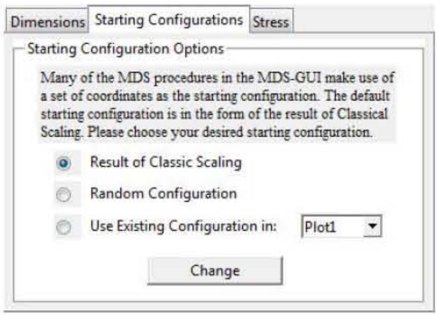
Figure 5.15 shows the three tabs of *MDS Options*, which is recalled through the *Multivariate Tools* menu. The purpose of *MDS Options* is to provide key adjustments to the MDS procedure that effect the output in a substantial way.

5.3.6.1 Dimensions Tab

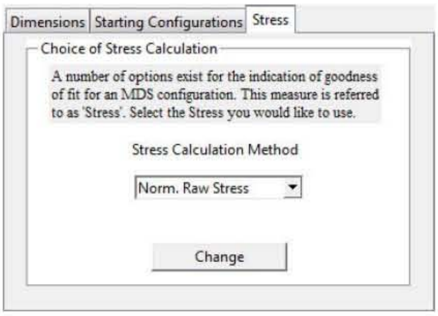
The first tab, *Dimensions*, adjusts the user defined p for all subsequent MDS procedures. The selection is possible to the user by use of a *tcltk “ComboBox”* which defaults to $p = 2$. When a new data set is added, the drop-box is populated with the entries being $1, 2, ..., n-1$. In the event that $p = 1$ is selected, that is some form of Unidimensional Scaling is required, the information box displayed in Figure 5.16 is produced which alerts the user to the tendency of Unidimensional Scaling to produce local minima. If the user still wishes to set p=1, they may continue.



(a) Dimensions Tab



(b) Starting Configuration Tab



(c) Stress Tab

Figure 5.15: MDS Options Menu

5.3.6.2 Starting Configurations Tab

The *Starting Configurations* tab (Figure 5.15(b)) provides the user the option to change what starting configuration is used in the MDS procedures (where starting configuration is relevant). Three options are provided, being: the $n \times p$ result of Classical Scaling on data; a random configuration, where the $n \times p$ matrix is uniformly distributed and doubly centered around the origin; and finally a configuration in any of the five main plotting tabs may be set as the starting configuration for all subsequent procedures. Therefore, the user may use, say, the result of a Sammon Mapping procedure as the starting configuration for a SMACOF procedure.

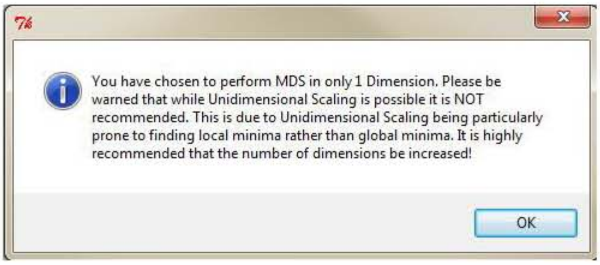


Figure 5.16: *p=1* Warning

5.3.6.3 Stress Tab

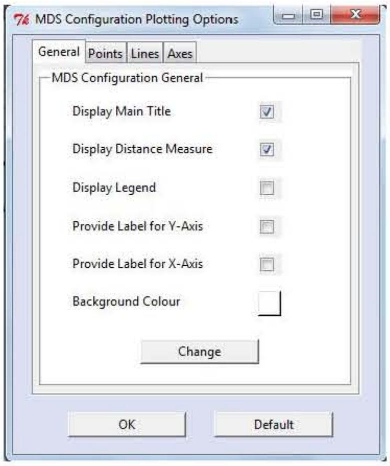
The final tab of *MDS Options* is called *Stress* and controls the measure of stress used to assess the goodness-of-fit of all configurations. It should be noted that this does not affect the loss function used within each MDS functions, as each method is usually defined by their specific loss function. The stress method chosen here simply defines how the accuracy of the final configurations are measured, in order to compare accuracy of configurations in absolute terms, as they need to compared on an identical scale. Options for this include: Normalised Raw Stress, STRESS1, STRESS2 and Pearson's Correlation Coefficient. Details on each of these can be found in Section 2.4.

5.3.7 Plot Option Menus

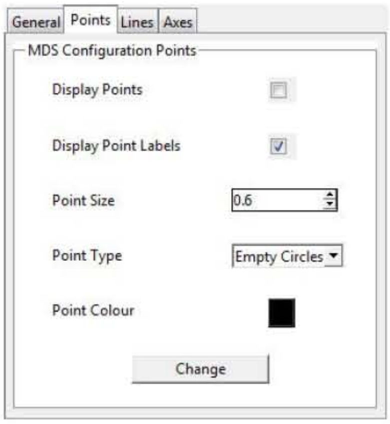
Each plotting area found in the MDS-GUI has a Plot Options menu, which may be accessed via a right click of the plot and selecting the *Plot Options* option. These areas include: Configuration Plot, Shepard Plot, Stress Plot(s), Scree Plot, Zoomed Plot, Procrustes Plot, Static 3D plot and RGL 3D Plot. Each of these menu's are set out in a similar way with only slight differences depending on the nature of the plot itself. The menu associated with the main plotting area(s) will be used for demonstrative purposes. Four tabs exist on the majority of the menus.

5.3.7.1 General Tab

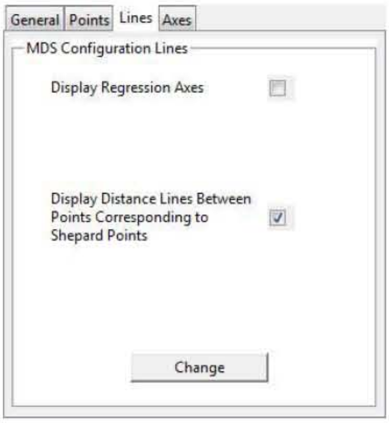
The *General* tab of the plot menu deals with overall settings of the plotting area. The first three options, *Display Main Title*, *Display Distance Measure*



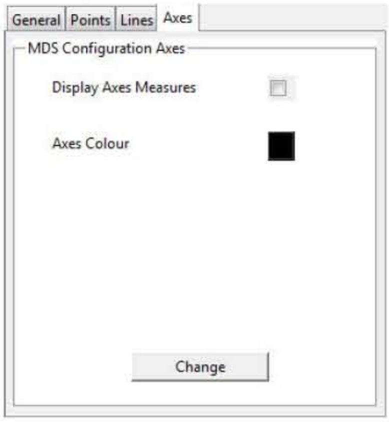
(a) General Tab



(b) Points Tab



(c) Lines Tab



(d) Axes Tab

Figure 5.17: Plot Options Menu

and *Display Legend* all toggle their appropriate settings on and off with no adjustability. Selecting to display the axes titles however opens a *tcltk* text box which prompts the user to name the axes as they see fit. This option might be appropriate, for example, when geographical locations have been plotted and the configuration rotated such that the dimensions are North-South and West-East. Finally, *Background Colour* allows the user to change the primary colour of the plot.

5.3.7.2 Points Tab

The visual effects of the n points making up the configuration are controlled by the *Points* tab. Each point may either be displayed as a point without a label, a label without a point or a point with the label above it (above is default. This location may be changed in the *General Settings* menu). Following this, the size, shape and colour of the point may be altered. Adjusting the shape applies to when the point is displayed, and the possibilities include: filled circles, empty circles, filled squares, empty squares, filled boxes, empty boxes and crosses.

5.3.7.3 Lines Tab

Lines may be added to the configuration plots in two different ways. The first is the variable axes (Section 5.4.12) and the second are the distance measures added via the Shepard Diagram (Section 5.4.3). The *Lines* tab toggles whether these lines are displayed or not with resetting their settings.

5.3.7.4 Axes Tab

The measurement indicators on the axes of the plotting area for the configuration are usually regarded as irrelevant. This is due to the fact that only the relative distances between points is useful which may be observed visually. This being said, the option to add the numerical axes measures is available to the user if that output is desirable to them. The colour of the axes (and measurements if activated) may also be altered.

5.4 Overview of Features

The MDS-GUI incorporates a large variety of features and applications of tools in order to cater for a range of a user's Multidimensional Scaling interpretation needs. This Section will outline and demonstrate the various features of the software. For the purposes of demonstration, the *skulls* data set, already used in Section 2.10.1 and described in Appendix A, will be used throughout the Section.

5.4.1 Plotting Tab Features

Five individual MDS configuration plotting areas exist within the MDS-GUI, as described by Area Two in Section 5.1.1. These areas allow for up to five different MDS procedures (when p=2) to be portrayed under different conditions in order to graphically observe any differences that may exist between them. The MDS-GUI defaults with *Plot1* in focus. Changing to the second plotting area requires the *Plot2* tab be selected, and so on. Upon changing the active plotting area, the new configuration plot area comes into focus in the *Main Plotting Area*. In addition, the Shepard Plot, Stress Plots and Scree Plot, all in the *Secondary Plotting Area*, are changed to correspond to the active area. This means that when *Plot1* is selected the Shepard Plot (and others) correspond to the *Plot1* configuration, etc.

The five main plotting areas are set up in such a way that they are completely independent and are governed by their own individual settings. Each area has its own separate *Main Plot Menu*, accessed by a right click with any changes made with this menu being isolated to the specific plot. Similarly, any visual and graphical alterations made using the *Plot Options* menu, are localised. This means that each plot can be completely user specified so that all output is adequately distinguishable.

Tab	Measure	MDS	Dims	NormRawStres	Plot.Dims	Tolerence	Iterations
Plot1	Minkowski.Metric	M.Smac.Sym	2	0.048	1&2	1e-05	109
Plot2	CityBlock.Metric	Kruskal	2	0.027	1&2	1e-05	35
Plot3	Wave-Hedges	ClasScal	2	0.734	1&2	-	-
Plot4	CityBlock.Metric	ClasScal	2	0.12	1&2	-	-
Plot5	Euclidean.Distance	Sammon	2	0.034	1&2	1e-05	30
Stat 3D	-	-	-	-	-	-	-
RGL 3D	-	-	-	-	-	-	-

Figure 5.18: Configuration Table

The *Configuration Table* located in the table Section of the MDS-GUI summarises all relevant information of each of the five plotting areas. This allows the user to make direct comparisons from a numerical point of view. Every time an MDS procedure is performed, the information is updated on the table in the row corresponding to the active plotting area. The table (Figure 5.18) stores the following information: The distance metric used to calculate the $\mathbf{\Delta}$ dissimilarity matrix (when relevant); the type of MDS performed; p; the stress value of the final configuration (adjustable in the *MDS Options* menu); the dimensions being plotted (when $p \geq 3$); the tolerance applied to the procedure; and finally the number of iterations to find convergence. It should be noted that the value reported in the stress column will only apply when the MDS type is Classical Scaling, Least Squares Scaling, Metric SMACOF, Non-Metric SMACOF, Kruskal's Analysis and Sammon Mapping.

5.4.2 $p \geq 2$

The default number of output dimensions, p, is two in the MDS-GUI. There are however allowances and capabilities to have p equal to any dimensions from 1 up to $n - 1$, where n is the number of objects in the data (See the *Dimensionality* Section of Chapter 2 for details on p). The options for p are available in the *MDS Options* menu.

5.4.2.1 Three Dimensions

The MDS-GUI has two separate means of portraying MDS configurations in three dimensions, that is when $p = 3$. Upon selection of an MDS method when $p = 3$ is set, the options window shown in Figure 5.19 is presented, prompting the user to choose from either plotting a static plot or an rgl plot (or both). The two possible outputs are displayed as (a) and (b) respectively in Figure 5.20.

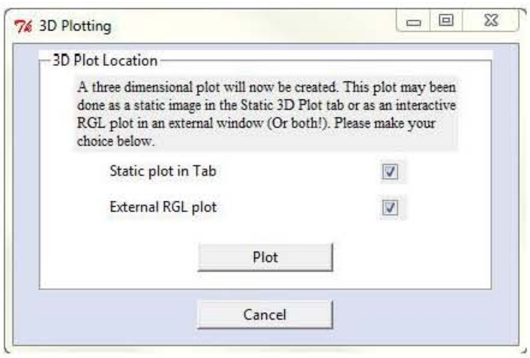
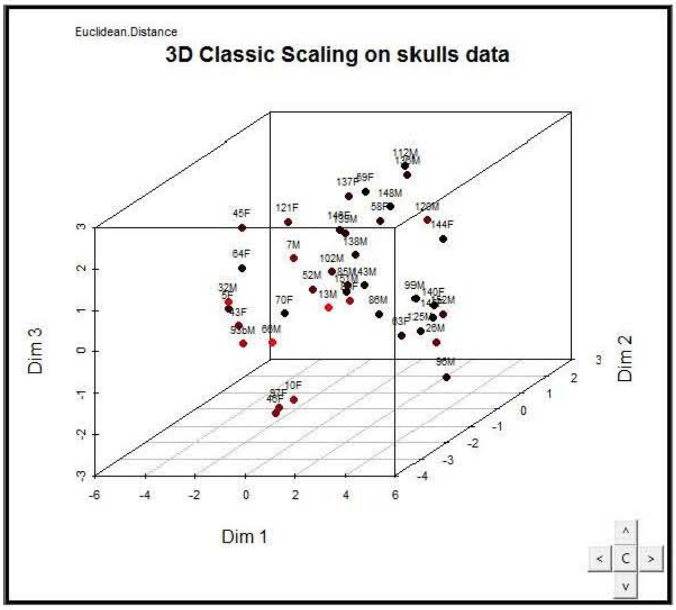
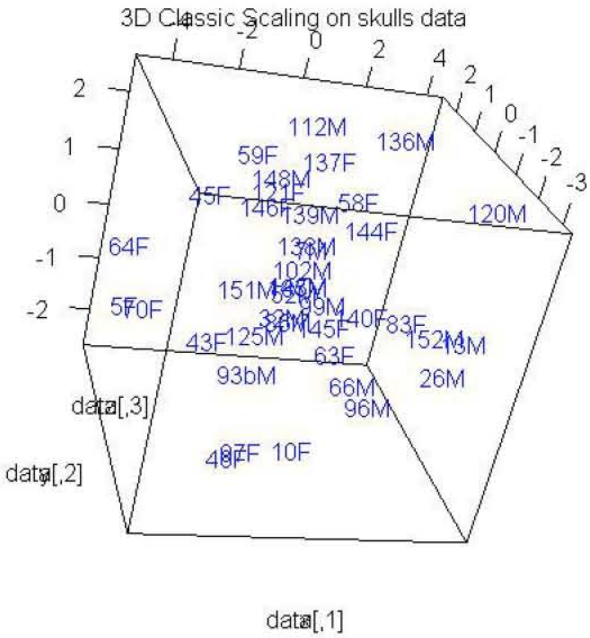


Figure 5.19: Three Dimensions Options



(a) Static 3D Plot



(b) RGL 3D Plot

Figure 5.20: 3D Plotting

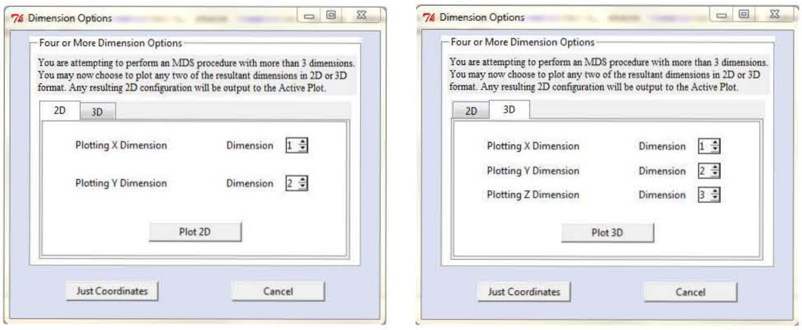
The Static 3D plot is a two dimensional depiction of the three dimensional configuration and is displayed in the *Static 3D Plot* tab of the main plotting area. There are two features of this plot that assist in the three dimensional visualisation. The first is the option to highlight the points with regards to their depth in the plot. Therefore, using Figure 5.20(a) as an example, the points with the least depth in the third dimension are coloured increasingly red, while those that are deep are coloured increasingly black. In addition, the arrow buttons at the bottom right of the configuration allow for rotation of the points both horizontally and vertically, which when clicking continuously provides a satisfying three dimensional depiction.

The RGL plot result (Figure 5.20(b)) is a more sophisticated and robust three dimensional tool. Since the function calls upon *R*'s built in rgl functionality, the result is separate from the MDS-GUI. The resulting RGL window is produced within the console when using *RGui* and is opened as a separate window when using *RStudio*. The impressive software allows the user to use their mouse cursor to rotate the plot in any direction with a simple dragging action. In addition, with the use of the mouse's wheel (or holding the right click button), the user may zoom in and out of the configuration at will in order to make detailed three dimensional observations of the MDS configuration of points.

5.4.2.2 More Than Three Dimensions

Upon performing an MDS procedure when $p \geq 3$, the options box, shown in Figure 5.21 is produced. Due to the fact that it is not possible to visually depict a figuration of more than three dimensions, the user is prompted to make a choice of an alternative means of producing their output.

Three different options exist for this scenario. The user may either, plot any two of the p dimensions in two dimensional space (Figure 5.21(a)), plot any three of the p dimensions in three dimensional space (Figure 5.21(b)), or not produce any plot and simply output the $n \times p$ $\mathbf{X}$ matrix of coordinates using the *tcltk* matrix editor (Figure 5.22).



(a) Large Dimensions: 2D Options (b) Large Dimensions: 3D Options

Figure 5.21: Large Dimensions Options

5.4.3 Shepard Plot

The Shepard Plot is housed in the first of the tabs of the *Secondary Plotting Area* and is thus the focused tab by default. Figure 5.23 shows the MDS-GUI after Metric SMACOF is performed. This scenario will be used to demonstrate the features associated with the Shepard Diagram. Details of the Shepard Plot can be found in Section 2.6.2, however a brief summary is that each point of the plot represents the distance between two of the n objects. The Y-Axis represents d, the MDS generated Ordination Distance, while the X-Axis represents δ, the Observed Dissimilarity distance.

The Shepard Plot defaults with dark points and the transformation curve plotted in red (identity function for Metric MDS methods and isotonic regression when Non-Metric MDS methods are used). This step curve gives a visual depiction of the isotonic regression transformation used during the Non-Metric MDS procedures. The plot will automatically change whenever the main plotting tab is changed and correspond to the MDS configuration displayed. However, the option is available to the user to observe a Shepard Plot relating to a non-active configuration tab for comparative purposes. This is achieved with the drop down menu located at the top left of the Shepard Plot area.

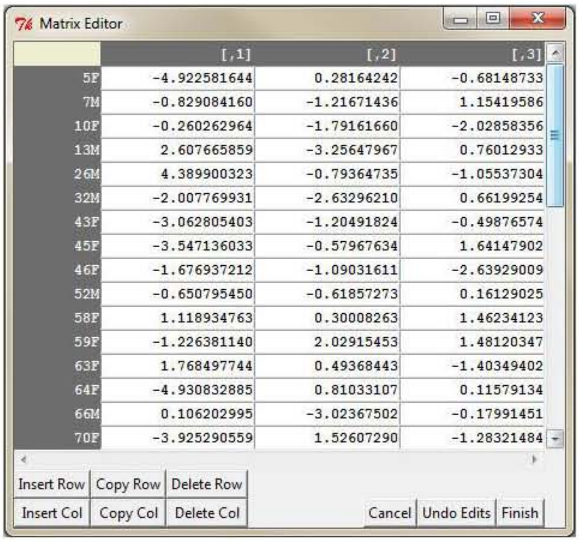


Figure 5.22: Matrix Editor

5.4.3.1 Shepard Point Labeling

With each point on the plot representing a specific data point pairing, it is useful to visually identify and associate the relationships between the Shepard Plot and the MDS Configuration Plot. This is done with the use of concurrently labeling a point of the Shepard Plot and have the pairing mapped on the Configuration Plot. The most direct method of performing this task is through simply left clicking a point on the Shepard Plot. Upon this selection the chosen point will be colour coded and labeled with the two corresponding point names. All non selected points at this point are changed to a subdued shade of light gray which, while still visible, are light enough to be unobtrusive in reading the highlighted point labels. Concurrently, a line is drawn on the Configuration Plot between the two points associated with the selected Shepard point. This line is in the same colour as the highlighted point, allowing the user to easily identify which line is associated with which point when multiple points have been labeled and thus multiple lines have been drawn. Figure 5.24 shows the result of four labeled points. It is useful to note that the Y-Axis coordinate of each point demonstrates the distance shown on the MDS configuration. Therefore, although the red and light blue points have the same X coordinate, the red line is longer than the blue

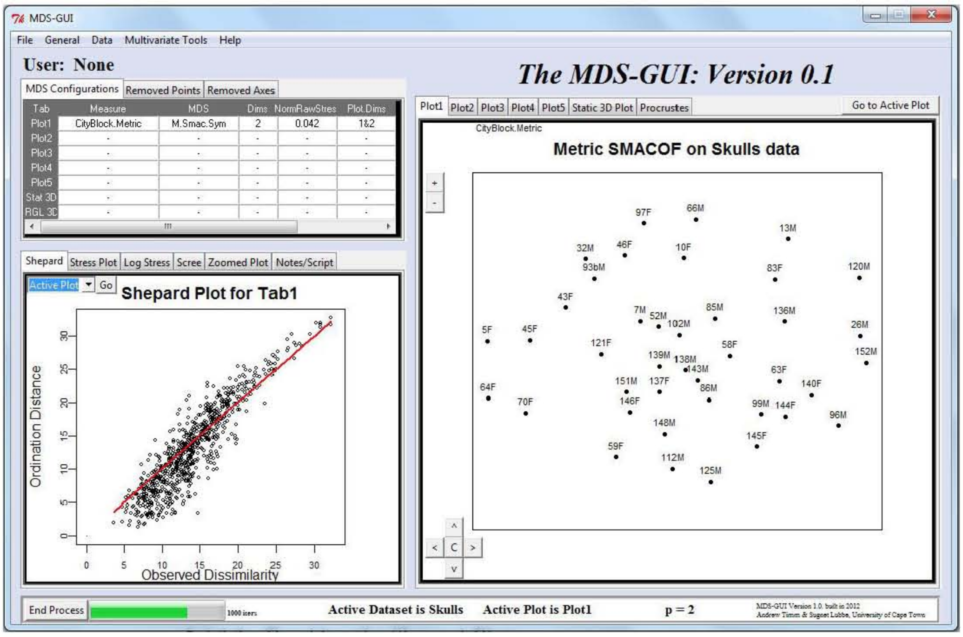


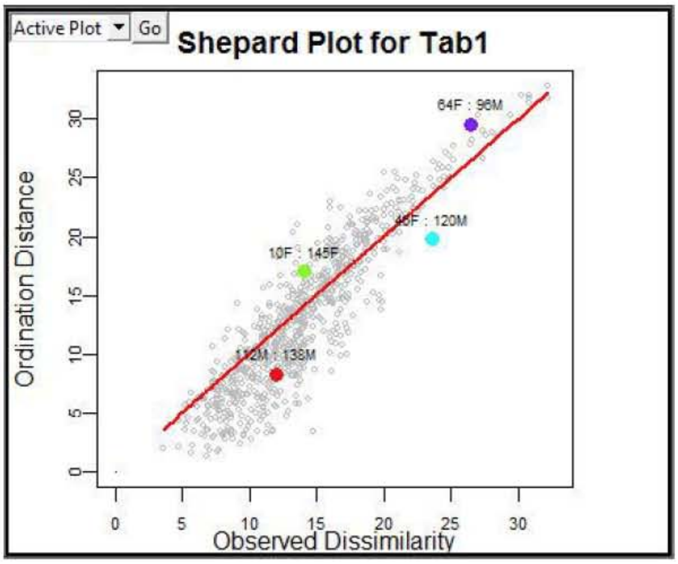
Figure 5.23: MDS-GUI: Displaying Shepard Plot

line.

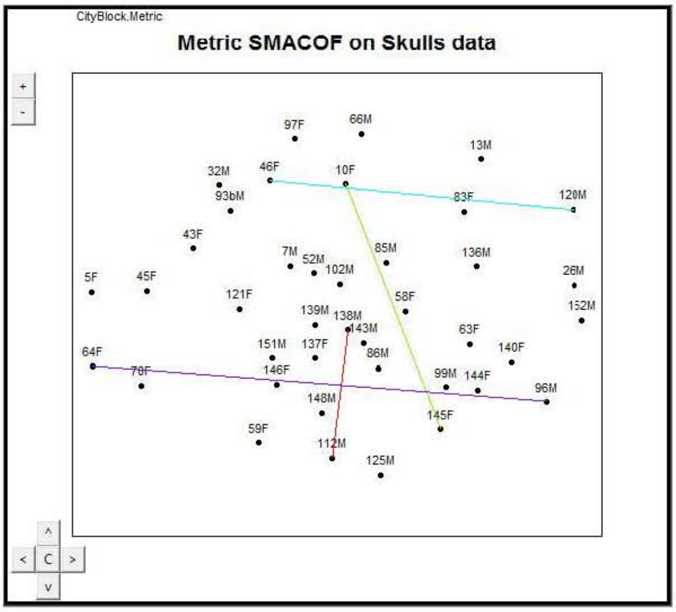
The second method for labeling Shepard Points is through the *Shepard Point Label* pop-out window (Figure 5.25). This is accessed through the right click menu of the Shepard Plot area. This method allows the user to locate a point on the Shepard Plot that represents a specific pairing. This is of use when the researcher is interested in how well specific pairings were represented in the MDS process. The selection tool for both points is a drop down menu with editable text box. The user may then either find the points in the automatically populated drop down list, or alternatively type the name of the points.

5.4.3.2 Brushing the Shepard Plot

The term "brushing" is often used, in the context of computer software, to refer to the act of selecting multiple items with a mouse cursor by holding a click of the mouse and drawing a rectangle around the points. The opposite corners of the box are selected as the points where the click was made and the point where it was released respectively. This method has been



(a) Labeled Shepard Points: Shepard Plot



(b) Labeled Shepard Points: Configuration

Figure 5.24: Labeled Shepard Points

incorporated into the Shepard Plot allowing the user to label entire groups of points at a time. This feature is particularly useful in identifying the objects associated with groups of outlying Shepard Points. Labeled points

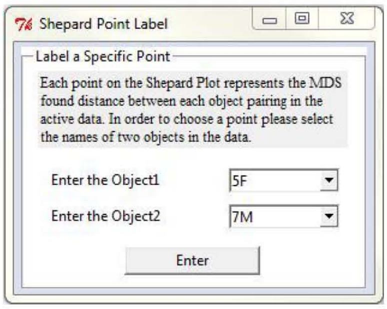
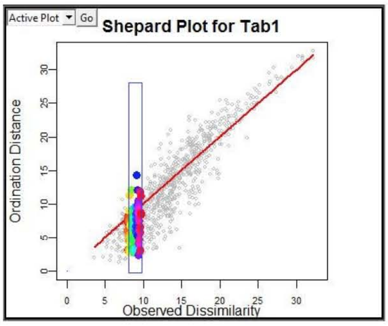
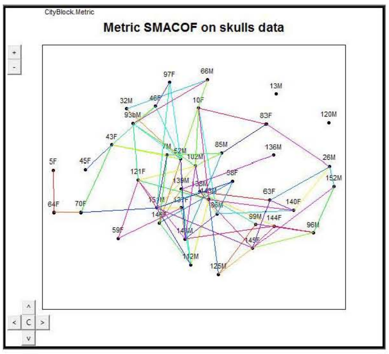


Figure 5.25: MDS-GUI: Specific Shepard Point Label

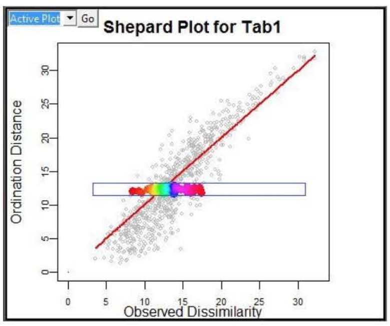
are similarly removed using brushing, as selecting an area without points will clear all labels from the plot. Due to crowding of labels, only ten points on the plot will display the point labels at a time, thereafter only the highlighting of points takes place.



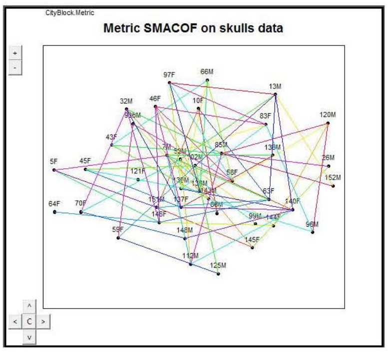
(a) Vertical Brushing: Shepard



(b) Vertical Brushing: Configuration



(c) Horizontal Brushing: Shepard



(d) Horizontal Brushing: Configuration

Figure 5.26: Shepard Point Brushing

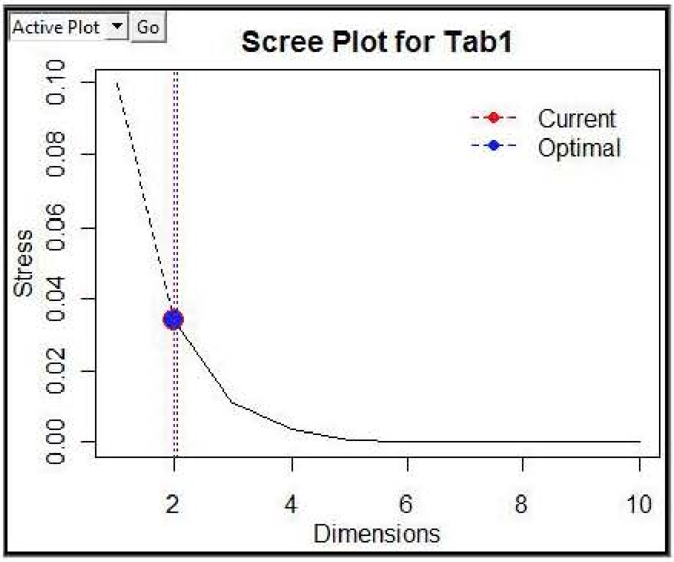
The brushing feature may also be used in demonstrating the interpretation of the Shepard Plot. Figure 5.26 shows the effect of brushing the Shepard Plot in two separate ways. The first is selecting a group of points within a narrow vertical column of the plot. The results of this are depicted in Figures 5.26(a) and 5.26(b). These points all have similar X-values but have a range of Y-values. The highlighted segment thus represents a group of points that should have the same distance, but have been assigned different distances by the MDS procedure. The line lengths in 5.26(b) are all different. The alternative scenario is shown in Figures 5.26(c) and 5.26(d), which is the effect of selecting a group of points along a narrow horizontal row on the plot. These points all have similar Y-values but differing X-values. This means that their resulting MDS distances are the same while their observed dissimilarity distances are different. Figure 5.26(d) thus shows lengths that are all inaccurately similar.

5.4.4 Scree Plot

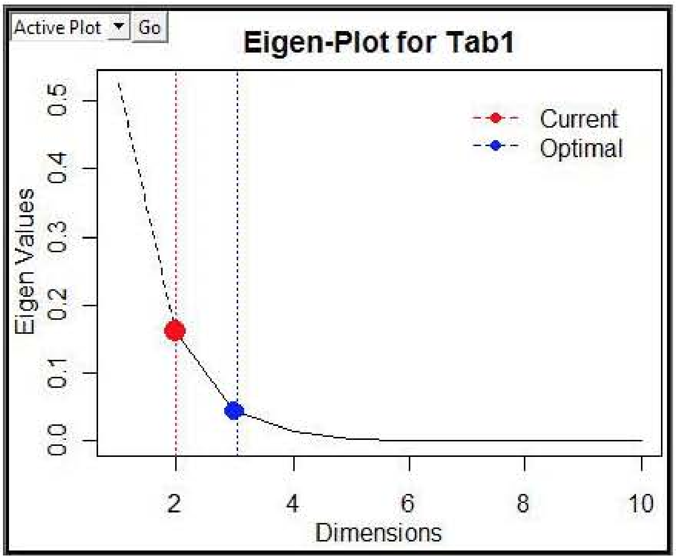
The Scree Plot, described in Section 2.6.1, is another diagnostic tool provided by the MDS-GUI. By default, a Scree Plot is generated every time an MDS procedure takes place for the first time on a set of data. This option may be deactivated if it is found that the computation time for the plot is too intensive. It is not uncommon for the calculations for the Scree Plot to be more time consuming, and this is due to the fact that although the MDS configuration output is produced only for a set p value, the Scree Plot requires stress calculations on every dimension from 1 to 10 (The 1:10 range is not a rule, but has found to be appropriate from a visual sense). The resulting Scree Plot can be found in Figure 5.27.

Two cases of Scree Plot exist within the MDS-GUI. The first is the typical plot of stress versus dimension, however in the case that Classical Scaling is used, an Eigen-Plot is produced. The nature of Classical Scaling is that the Stress of dimension r is equivalent to the Eigenvalue relating to the r^{th} Eigenvector. Thus, when Classical Scaling is performed, the Scree Plot plots eigenvalues versus dimension.

The plot itself has two points highlighted on it. The first identifies p,



(a) Standard Scree Plot



(b) Scree Plot: Classical Scaling

Figure 5.27: Scree Plot

the number of plotted dimensions, and labels it “Current”. The second point illustrates the calculated possible “Optimum” dimension to use for the specific data set. Section 2.6.1 describes the optimum dimension by the “kink” in the curve, and based on this logic the point is identified. The point at which the change in angle of the curve is greatest corresponds to the optimum location. For example, Figure 5.27(a) is the resulting Scree Plot from a Sammon Mapping procedure with p=2, and the optimum dimension is identified as two. Figure 5.27(b) however is the result of Classical Scaling with p=2 and the optimum dimension is identified as three. The display of these points may be toggled on and off per the users preference.

5.4.5 Iterations Observable

Most Multidimensional Scaling methods are reliant on iterative procedures. The nature of this is that at the end of every iteration, an $n \times p$ matrix of coordinates is produced. This means that, through plotting the coordinates after every iteration, it is possible to observe the minimisation procedure taking place. With current computing power, even basic personal computers are able to compute and display dozens of iterations per second, which results in an apparent fluid and continuous motion of the points in the configuration. The optimisation procedure is therefore animated. The Shepard Plot is also active throughout the iterations, and this is due to the fact that the input required for the plot is the dissimilarity matrix, $\mathbf{\Delta}$, and the $\mathbf{D}$ matrix derived from the coordinate matrix (and $\hat{\mathbf{D}}$ matrix when non-metric); both of which are available after each iteration. When the iterative changes are displayed on the Shepard Plot, the process is often slowed down considerably as more animation and therefore calculations are required per iteration. Preventative measures are in place to avoid unnecessary time wasting, including deactivating iterative Shepard Plot updates. Alternatively, the Shepard Plot will only update when the Shepard Plot tab is in focus on the MDS-GUI.

5.4.5.1 Stress Plots

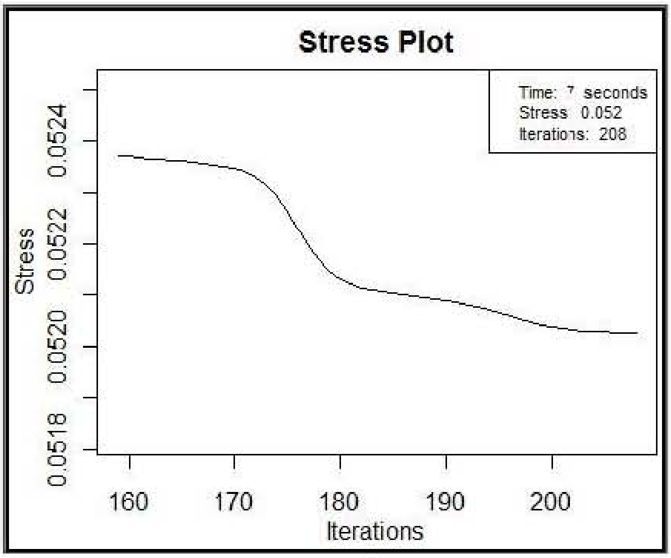
A ‘Stress Plot’, in the context of Multidimensional Scaling, is a plot of stress versus iterations and are useful for observing at which points in the process is stress most effected. The *Stress Plot* tab in the *Secondary Plotting Area* provides such a plot. In addition to the plot, a variation of the stress plot exists in the form of plotting the logged differences of stress versus iteration.

This rendition of the stress plot is found in the *Log Stress* tab. Figure 5.28 shows both the Stress Plot and Logged Differences Stress Plot for the same point in time of an MDS process. Both plots are also updated after every iteration (when the tab is in focus), allowing the user to observe the status of the stress throughout the procedure. It should be noticed that by definition, stress will only decrease as iterations progress, meaning that the slope of the Stress Plot is always negative. The slope of the logged difference stress plot will however be positive when the rate of decrease of stress increases, and the slope will be negative when the rate of decrease of stress decreases. The drop in stress at 170 iterations in Figure 5.28(a) is therefore matched by a peak at 170 iterations in Figure 5.28(b).

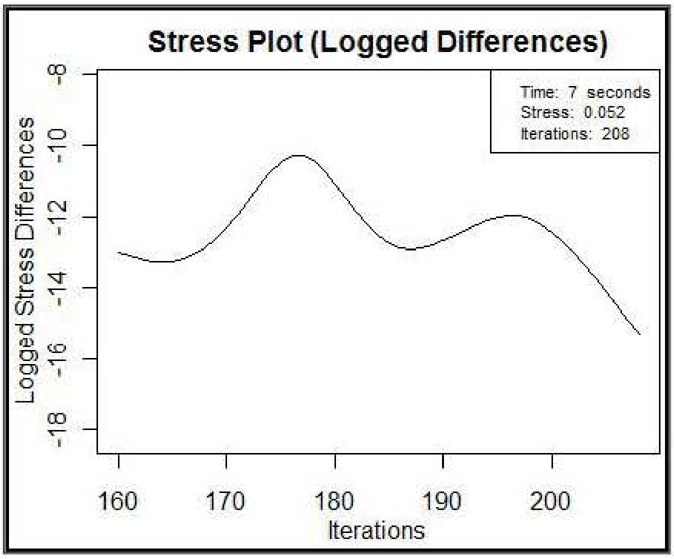
The box at the top right of both Stress Plots provides further information regarding the MDS process under way. The time in seconds, the current level of stress (measured according to the Stress method selected in *MDS Options*), and the number of iterations is displayed after each iteration. It should also be noticed that only the last 50 iterations of the process are provided on both plots as this allows a more accurate portrayal of the relative change in stress as iterations progress.

5.4.5.2 Progress Bar and End Process

The final tool of the MDS-GUI relating to the iterative nature of the procedures is the *Progress Bar* and *End Process* button (Figure 5.29) found in the *Information Pane*. The progress, in terms of percentage of iterations over maximum iterations, is tracked using the progress bar, with the bar being complete when maximum iterations have been reached. The button alongside the progress bar is available to the user to end any MDS process before it is complete. During procedures, the mouse cursor indicates that a process is under way and all menus are unavailable; therefore if a procedure is proving to be more time consuming than expected, the user may terminate the process and regain access to the interface once more.



(a) Stress Plot



(b) Stress Plot: Logged Differences

Figure 5.28: Stress Plots

5.4.6 Procrustes Analysis

The multivariate process known as Procrustes Analysis is discussed in Section 2.10.3. The MDS-GUI has built in Procrustes Analysis features which

Figure 5.29: Progress Bar and End Process Button

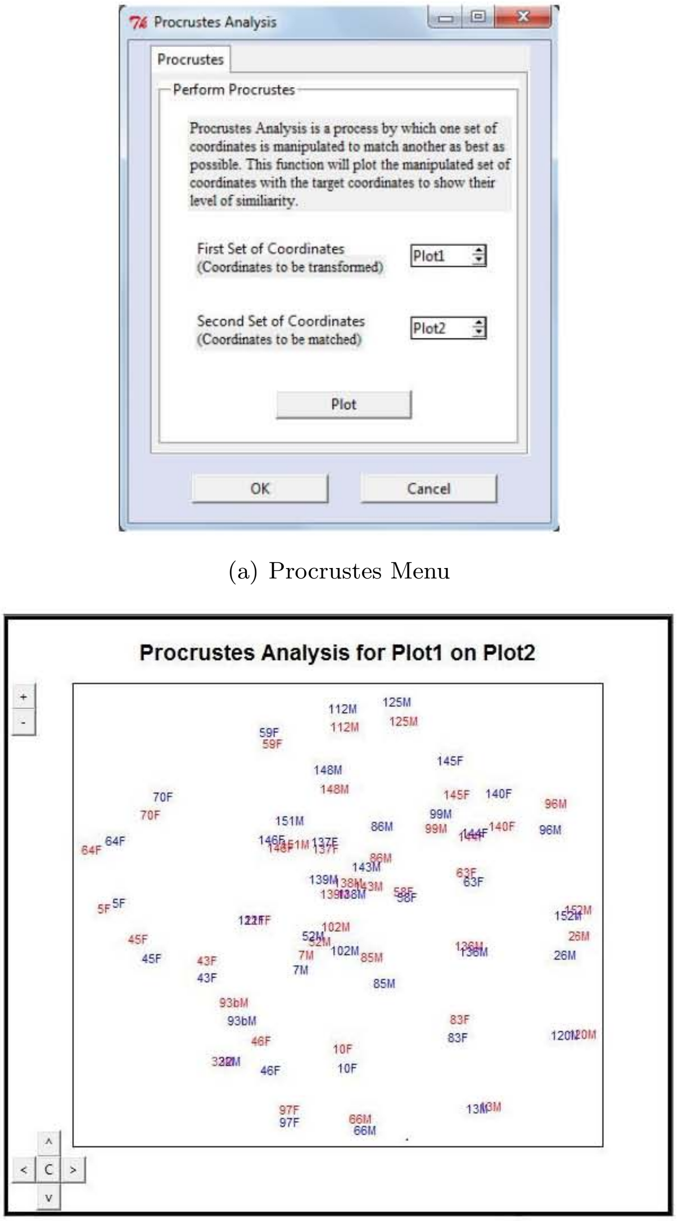
allow the comparison between any two 2D MDS based configuration. The *Procrustes Analysis* options window (Figure 5.30(a)) is called from the *Multivariate Tools-Topmenu*. The user is prompted to choose the two configurations in which to perform the analysis on. The options available are the configurations represented in any of the plotting tabs of the main plotting area. The resulting plot is then displayed in the Procrustes tab of the main plotting area. In the event that either, one or both of the plot areas are empty or the two selected configurations do not conform (that is $n_A \neq n_B$), an appropriate error message is returned and no analysis procedure is performed.

Figure 5.30 shows the example of Procrustes Analysis performed on the configurations from the first (red) and second (blue) main plotting tabs, which correspond to Kruskal's Analysis and Metric SMACOF respectively. Visual alterations may be made to the configuration, including adding an information legend, via the *Procrustes Plot Options* menu accessed via the right click within the *Procrustes* tab.

5.4.7 Point Labeling

A technique available to the user to highlight certain points on the configuration is through the use of labeling points individually. Two methods are provided by the MDS-GUI for individual point labeling. The first method is by the use of the mouse left click over a point. This will create the label above the point (or whichever position selected in *General Settings*). Figure 5.31 gives an example of a configuration (displaying only points) with four of the points labeled using the mouse cursor.

The second method is through the *Point Label* window (Figure 5.32) accessed via the right click menu of the plot. This window provides the user with the chance to select a single point to label, through either inputting the object name in the text box or selecting it from the drop down menu. This is particularly useful when a researcher is interested in the location of a



(a) Procrustes Menu

(b) Procrustes Configuration

Figure 5.30: Procrustes Analysis Example

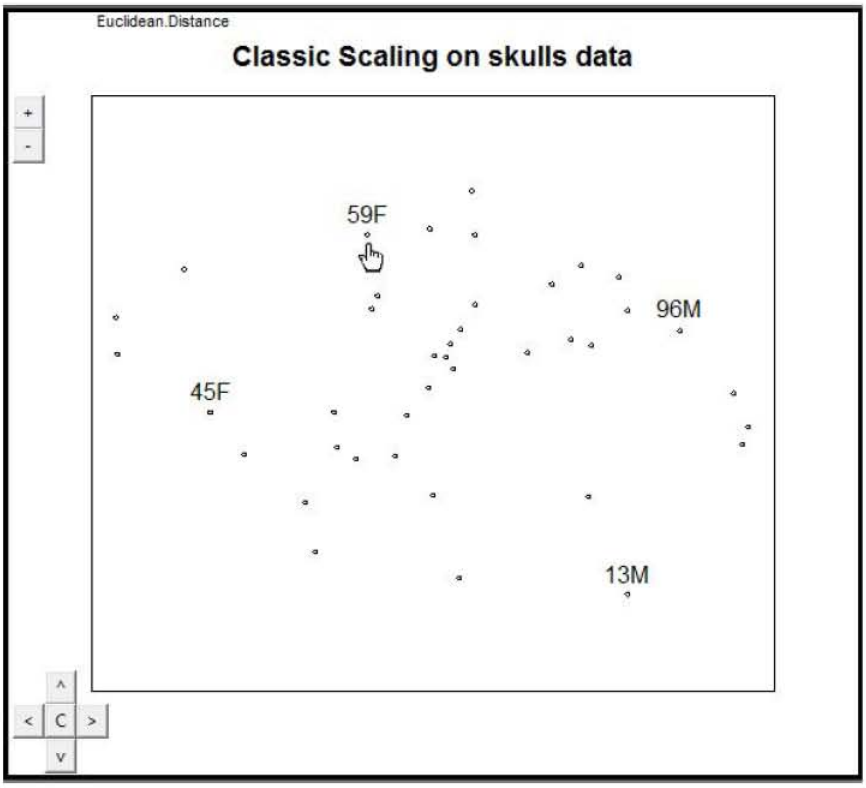


Figure 5.31: Label Point with Cursor

specific point and n is large enough to make visual identification of the point problematic. In either case of selection, all point labels are cleared through the right click menu and selecting the *Clear Added Point Labels* option.

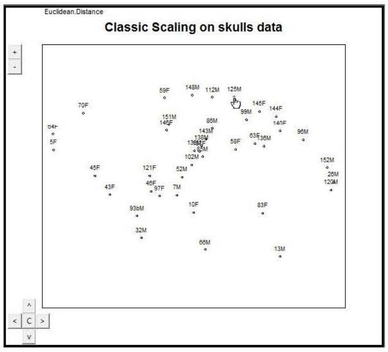
Figure 5.32: Label Specific Point

5.4.8 Manual Alterations of Configurations

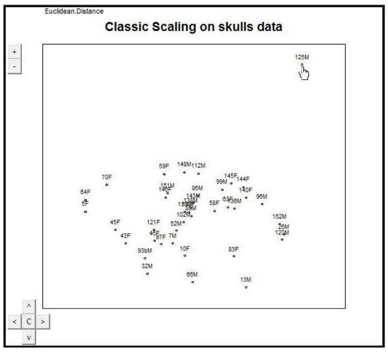
Any MDS configuration produced by the MDS-GUI is fully adjustable by the user, meaning that they are able to reallocate point coordinates and remove points. While the manipulated configuration loses all statistical meaning, there are benefits involved with having power over the output. The first is that with every frame movement of a point, the stress value for the configuration on the *MDS Configurations* table is updated automatically, meaning that altering the configuration will inform the user by what extent the change effects the overall stress. The Shepard Plot is also updated automatically. This kind of knowledge gives a user a better understanding of the makeup of stress and the configurations relationship with the Shepard plot. The second use for configuration manipulation is in the user defining an exact configuration that may be used as a starting configuration for a subsequent MDS procedure. The *Use Coordinates as Starting Configuration* option in the plot's right click menu automatically performs the same MDS procedure as the current plot, but with the altered configuration as an input.

Points are removed one at a time through the *Remove a Point* option in the right click menu. Once the option is selected, the user is prompted to select a point with their cursor. Once a point is removed, the left click function is reverted to point labeling and further point removals require the process repeated.

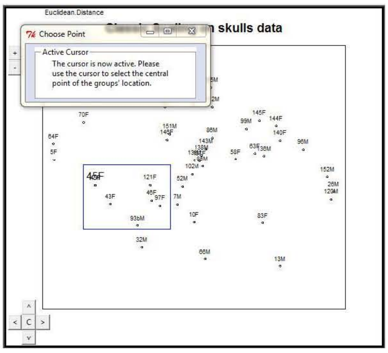
Two methods exist for the reallocation of point coordinates. The first is with the use of dragging the point with the mouse left button. Figure 5.33(a) and 5.33(b) show a point being relocated with the cursor. It should be noted that while a single click of the left mouse button is by default used for point labeling, holding the click is a separate operation on a computer's mouse, and the function associated with it is point relocation. The second method is to move multiple points in the configuration and preserve their internal positioning. This is achieved through 'brushing' with the cursor and the feature is selected with the *Relocate Group of Points* option from the right click menu. The user is first prompted to choose the group of points by drawing a box around them, as shown in Figure 5.33(c). With the group chosen, the user is then prompted to select the point on the mapping area that is to be the central point of the group. Figure 5.33(d) shows the relocated group with their relative positions preserved.



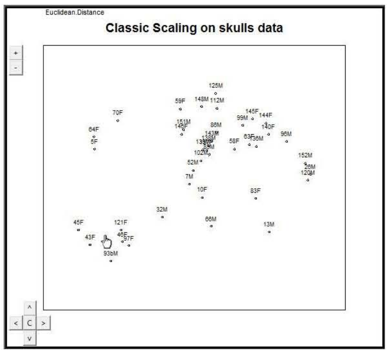
(a) Cursor Moving: Origin



(b) Cursor Moving: Final



(c) Group Moving: Origin



(d) Group Moving: Final

Figure 5.33: Moving Configuration Points

5.4.9 Zoom

A feature that is expected to be present in most mapping software's is the ability to zoom in and out of the plot in order to observe greater detail in different areas. The MDS-GUI provides two separate means for zooming. Manual zooming is available using the + and - icons at the top left of the main plotting area as shown by Figure 5.34 (Keyboard shortcuts using the '+' and '-' keys are also in place). Together with the *Repositioning Controls*, discussed in Section 5.4.10, one may focus on any point of the configuration, in whatever detail, with ease. This in no way effects the coordinates of the points, but merely adjusts the bounds of the scaffolding axes displayed on the plot. The alternative zooming options are of use when the data set is large and the zoom is required to be more advanced.

Figure 5.34: Manual Zoom Controls

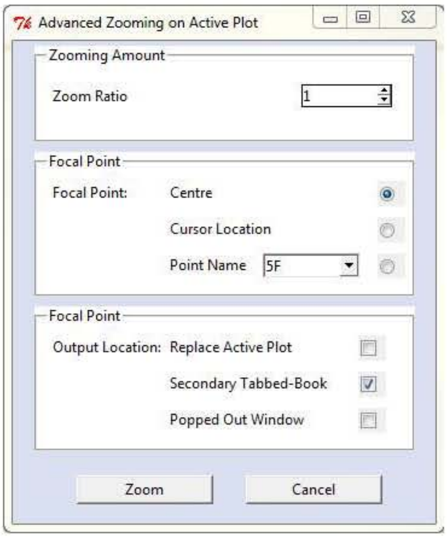
The advanced zooming procedure is controlled through the *Advanced Zoom Menu* (Figure 5.35(a)), accessible via the *Main Plot Menu*. The menu consists of three, user definable, components. The first pane of the menu prompts the user to select the extent of the zoom with the use of the *tcltk ComboBox*. Secondly, the *Focal Point* pane, requires the user to choose the point of origin of the new zoomed plot to be created. The three possible focal points options are: The origin of the original configuration; the location of the cursor upon the next left click of the user's mouse; and finally, any of the points in the configuration, chosen by an automatically populated *ComboBox* listing each of the object names. The third setting requires the user to select the location of the output. The three available options are: replacing the original configuration in the main plotting area; the *Zoomed Plot* located in the Secondary Plotting area; and as a popped-out plot.

Figure 5.35(b) shows the result of a 2× zoom, with a central point selected with the cursor and displayed in the *Zoomed Plot* tab of the *Secondary Plotting Area*.

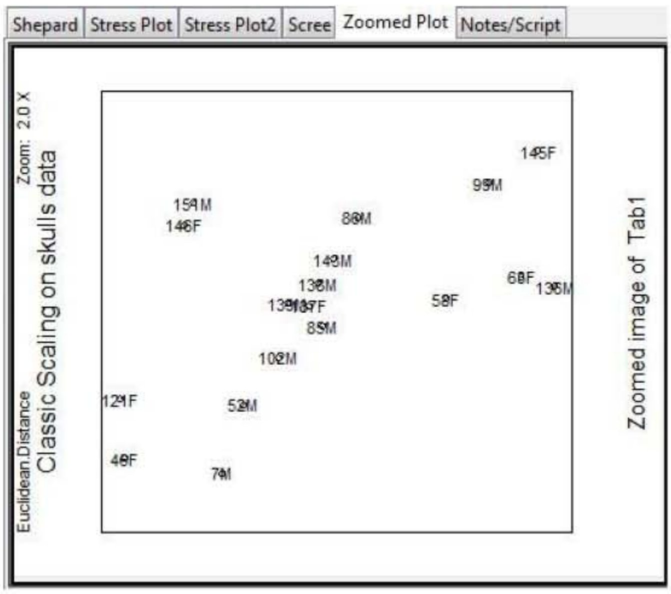
5.4.10 Configuration Orientation

The output configuration of an MDS procedure is produced in such a way that the orientation of the p axes is arbitrary. This is because it is only the relative distances between points that is of relevance. For this reason, the user of MDS based software are free to adjust the axes orientation of their output without limitations.

Figure 5.36(a) shows the menu window containing options to rotate the configuration points, or to reflect them about either axes. For rotation, the user is prompted to choose the direction of rotation (clockwise or anti-clockwise) and the degrees by which the rotation is made. For reflection, either reflection about the X-Axis or the Y-Axis (or both) need be chosen.

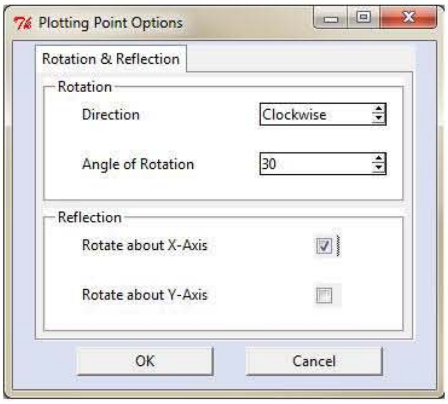


(a) Advanced Zoom Options

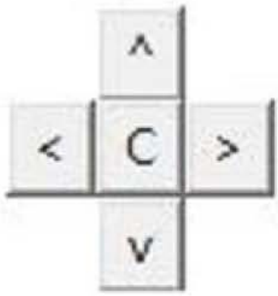


(b) Zoomed image in Secondary Plot tab

Figure 5.35: Zoom Options



(a) Rotation and Reflection Options



(b) Repositioning Controls

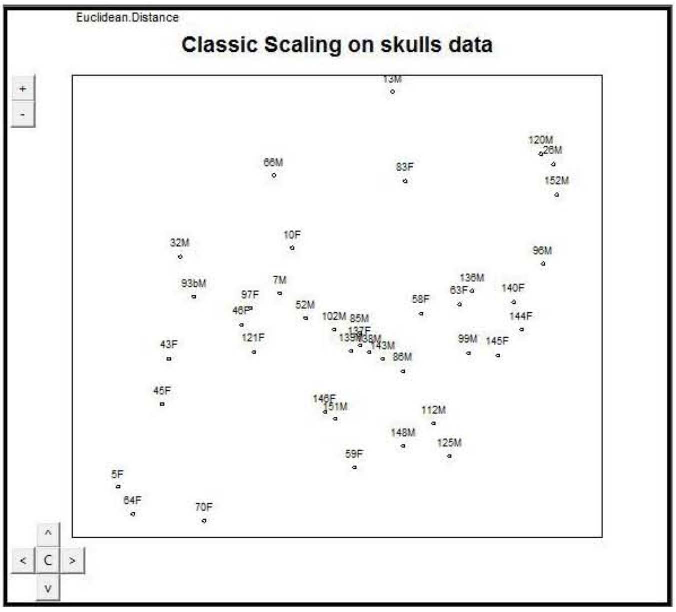
Figure 5.36: Configuration Orientation Controls

Operations of rotation and reflection may be performed in the same action. Figure 5.37(a) shows the result of a 30 degree clockwise rotation, accompanied by a reflection about the X-axis on the Classical Scaling result displayed in Figure 5.33(a).

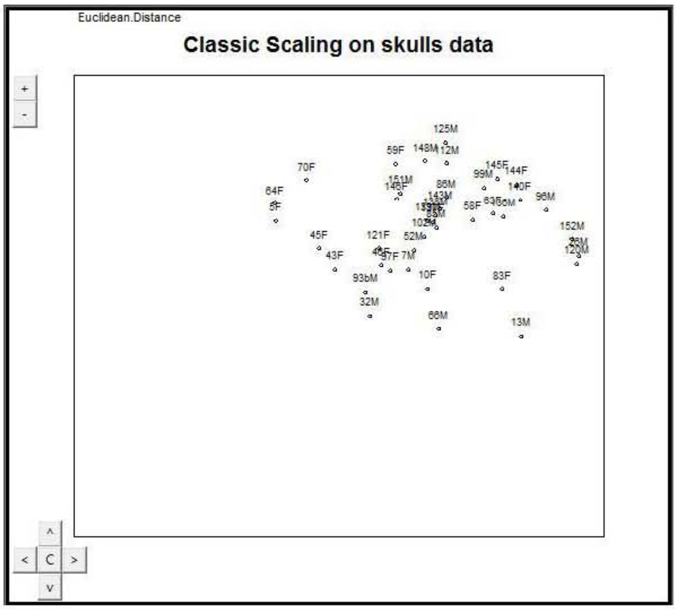
The arrow-like controls found on each of the main plotting tabs (Figure 5.36(b)) may be used to quickly and efficiently adjust the range of each axis of the plotting area. The effect of this is the appearance that the configuration is being moved from side to side and up and down. These controls, in conjunction with the manual zoom controls (Figure 5.34) give the user a high quality control of the focus of the configuration. Figure 5.37 shows the trivial result of zooming out, shifting the X-Axis to the left and the Y-Axis down. The *C* button in the center of the controls resets the configuration, that is, clears all manual zoom and shifted axes alterations.

5.4.11 Colour Options

The use of colour coding is a useful tool in the interpretation of ordination type mapping. It provides a means of immediately distinguishing different categories or points of interest. The MDS-GUI comes with a range of colour based features. The *New Active Dataset Options*, discussed in Section 5.3.1.3, requires the user to select a column of the uploaded data-file to be defined as a list of the object categories. Each category is automatically



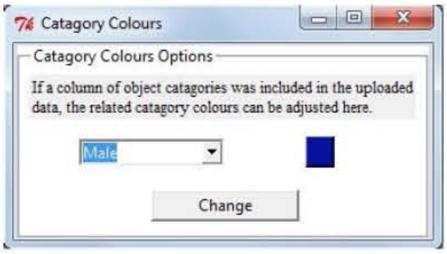
(a) Rotated and Reflected Configuration



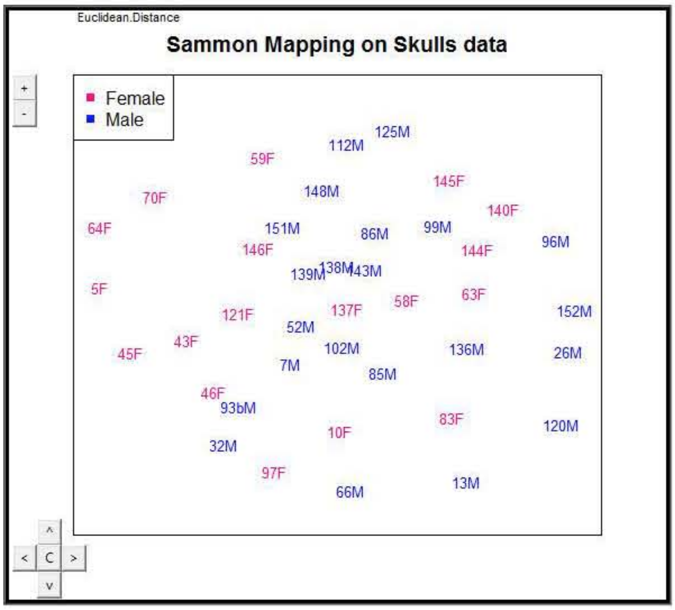
(b) Repositioned Plotting Axes

Figure 5.37: Configuration Orientation Changes

assigned a colour identifying the plotted points. While category colours are assigned automatically, the user is able to make adjustments. Figure 5.38(a) shows the *Category Colours* window, where the user may select the name of the category and then, by selecting the coloured box, adjust the colour. All colour alterations are reflected immediately.



(a) Colour Category: Options

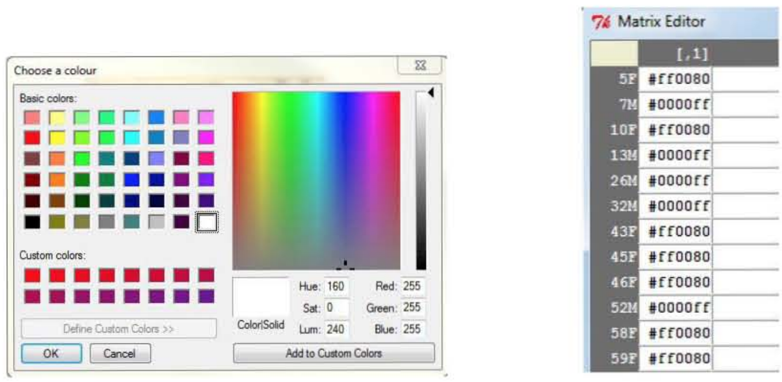


(b) Colour Category: Configuration

Figure 5.38: Colour Categories

The skulls data, for example, distinguishes between male and female skulls. A column is then added to the data file indicating the gender of the skull with either “Male” or “Female”. The user is then able to adjust the colours to the globally recognised colour for each gender, as indicated by Figure 5.38(b). In order to make the necessary colour changes, an operating system based colour editor is called by suitable *R* and *tcltk* commands. The *Microsoft Windows* example is given in Figure 5.39(a).

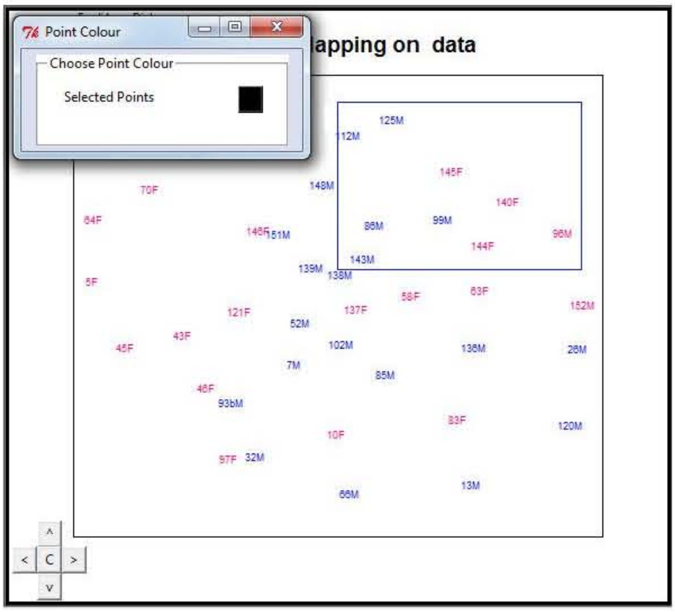
The *Data Colour Index* in the *Data Menu* calls a window like that shown in Figure 5.39(b). Each of the n objects is displayed and is accompanied by a six (sometimes eight) digit code. This sequence is the computer, *R* coded, tag for whichever colour is assigned to the object. Each element can be edited by the user, either in the form of a known coded sequence, or alternatively in one of the colour names recognised by *R*, e.g. "green".



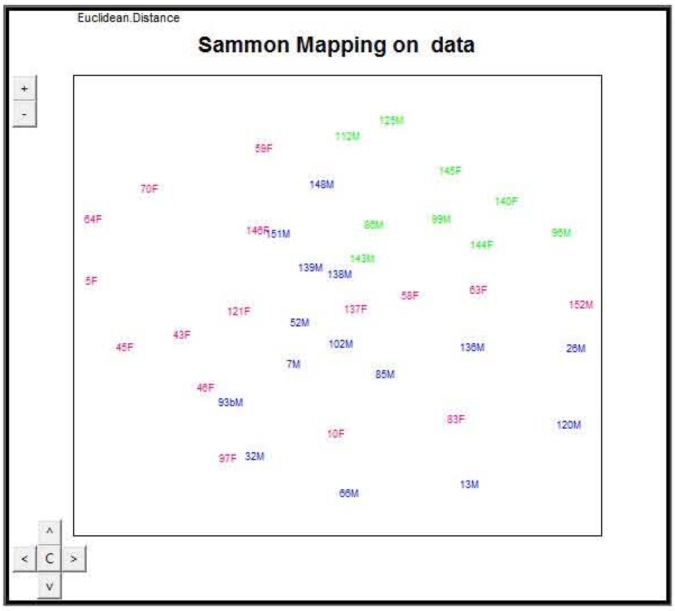
(a) Windows Colour Editor (b) Point Colour List

Figure 5.39: Colour Tools

The final method for altering colours of points is manually, with the mouse cursor on an already plotted MDS configuration. The brushing feature is called using the *Change Point Colour* command from the main plot right click menu. The user is prompted to draw a box around the points that are to have their colours altered, and then make the necessary colour adjustment. Figure 5.40 illustrates this process. It should be noted that changing colours manually limits the change to the specific tab that the change is made, whereas changing category colours is carried through all five plotting tabs. The *Default Point Colour* option then either returns all points to the default black colour, or, when applicable, to their category defined colours. When categories have been defined, a legend may be added to the top left of the configuration plotting area indicating the assignment of colour to each category. This legend is added via the relevant *Plot Options Menu*.



(a) Change Colour: Brushing



(b) Change Colour: Configuration

Figure 5.40: Change Point Colour

5.4.12 Variable Axes

The underlying axes, based on the variables found in the data, are discussed thoroughly in Section 2.10.2. These axes are usually defined by the m variables making up the data when the uploaded data is in the form the $n \times m$ $\mathbf{Z}$ matrix. When this is the case, (i.e. no similarity/dissimilarity matrix has been uploaded directly) the MDS-GUI is capable of offering graphical representations of the m variable axes. When applicable, the *Display Variable Axes* option in the right click menu of the main plotting area is activated, and the result of its selection will produce something similar to Figure 5.41. It will be noticed that each of the axes pass through the origin, which by default is at the center of the plotting area. Each of the m lines of axes is assigned its own colour and is labeled by its corresponding variable name. In addition, the value markers are present on each of the axes, indicating the progression of magnitude of the variable. The option to display variable axes is a setting unique to each of the five plotting areas, and selection of this option will be held throughout all future MDS plots performed in that area. A visually pleasing feature of this is that the movement of the axes may be observed throughout all iterative updates in future MDS processes. The user is able to view the relation of each axes in accordance with the configuration as the procedure progresses and eventually converges.

It is not uncommon for a researcher to be interested in only a selection of the m variables axes; in which case, displaying all of them may be non-ideal. The *Remove Axes of Variable(s)* option of the right click menu is available when axes display is activated. The options window, shown in Figure 5.42(a), is produced. A drop down list of each of the variable names is provided in this menu, from where the researcher may remove all undesirable entries one at a time. The configuration with five of the twelve original variable axes is shown in Figure 5.42(b).

5.4.13 Removed Points and Axes

Sections 5.4.8 and 5.4.12 made reference to the ability to remove points from the configuration and variable axes from display respectively. In both cases, the information of the removed items are preserved by the MDS-GUI and available in the *Removed Points* (Figure 5.43(a)) and *Removed Axes* (Figure 5.43(b)) tables. The *Removed Points* table lists each of the objects that have been removed from the configuration. The colour of each item

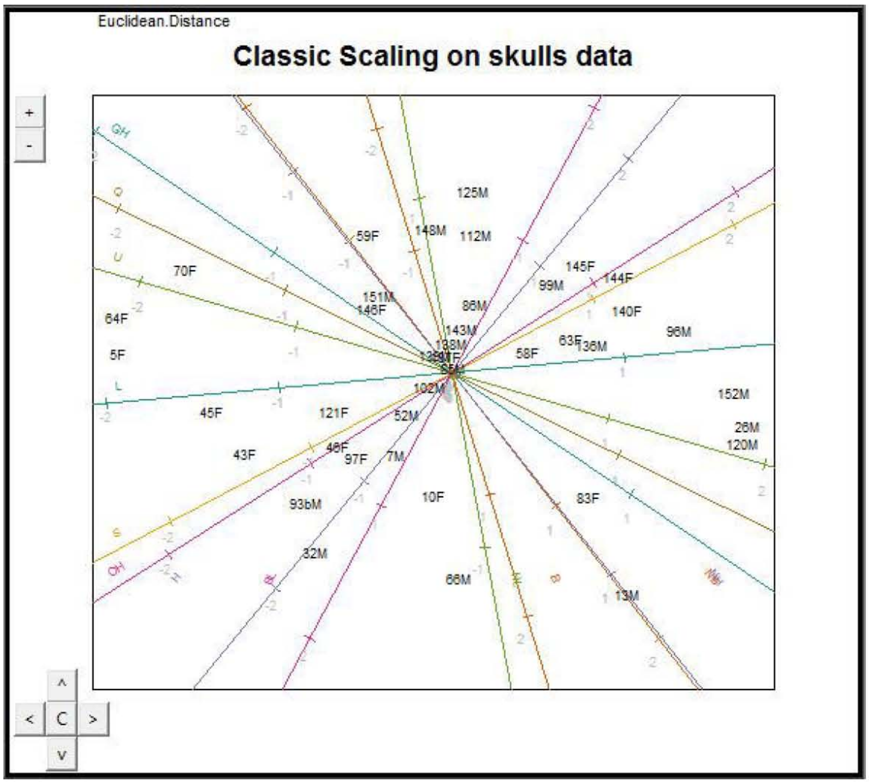
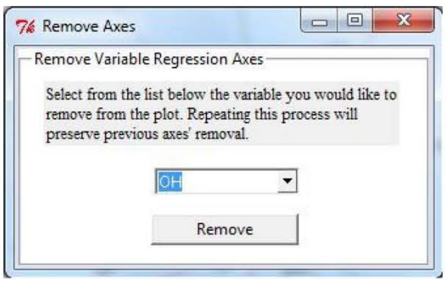
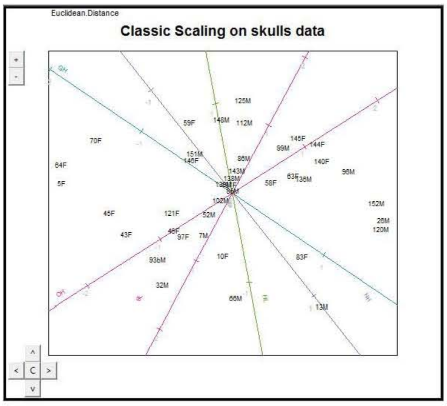


Figure 5.41: Display Variable Axes



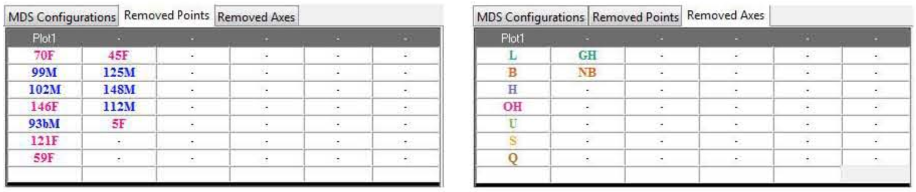
(a) Remove Axes Options



(b) Removed Variable Axes

Figure 5.42: Variable Axes Removal

in the table is the same as the object was before being removed from the MDS configuration. The example therefore shows all male objects being in blue font and all females in pink. Similarly, each of the m variable axes is allocated an individual colour, which are preserved in the list of removed axes, with each entry being the variable name relating to the axis. It is important to notice that at the top left of both example tables, *Plot1* can be found, meaning that these tables relate specifically to the first plotting area. When *Plot2* is selected to focus, the *Removed* tables will update and show the information relating to second plot, and so on.



(a) Removed Points Table (b) Removed Axes Table

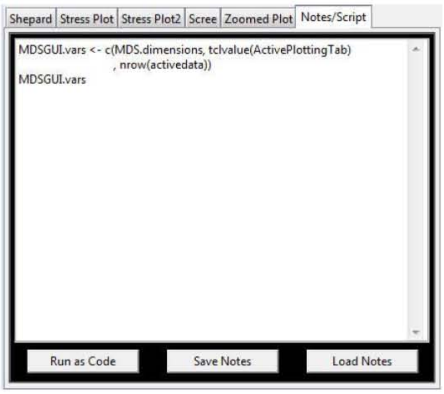
Figure 5.43: Removed Item Tables

Each removed item may be returned to the configuration by simply right clicking the relevant table entry and selecting the *Replace Point in Active Cell* option. Variable axes being returned are automatically placed in their correct position. When points are removed from the configuration however, their coordinate information is lost. Returned points therefore are relocated in the top-left corner of the plotting area, where the user may reposition them as they please.

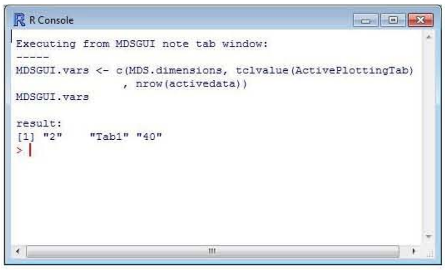
5.4.14 Notes/Script

The tab in the secondary plotting area named *Notes/Script* serves two purposes within the MDS-GUI. The first is a note making service available to the user where it acts as a simple text input location. The area comes complete with copy and pasting functionality, either through a right click menu or the keyboard shortcuts Ctrl-C and Ctrl-V. Any session of notes made by the user may be saved to an external location using the convenient *Save Notes* button, and can load any previous sessions with the *Load Notes* button.

The second feature is the interface that has been customised between



(a) R Scripting in MDS-GUI



(b) RGui Code Output

Figure 5.44: Notes-Script Tab

the tab and the active *R*-Environment. This link allows the user to treat the area as a scripting location, whereby they may write their *R* code and run it directly from the MDS-GUI. Figure 5.44(a) shows a simple *R* code script in the tab which, when run, will list the current value of p, the text component of the *Tcl* object controlling the active main plotting tab, and the number of rows of the uploaded data. Using the *Run as Code* button produces the output shown in Figure 5.44(b) in the *RGui* console.

5.4.15 Save and Load Workspace

The *Save Workspace* and *Load Workspace* functions were incorporated into the MDS-GUI in order to make the process of reproducing results as easy as possible for the user. The MDS-GUI was designed in such away that the user has a great deal of control over the particulars of the settings of the software and output of the results. If a researcher makes regular use of the program and has come to find a certain set of settings most appropriate or appealing to them, they may save their workspace in such a way that when returning to the MDS-GUI they may simply load their personal workspace back and regain their style without having to take time resetting it. In addition to the settings being restored, all plots and tabular information are stored as well and replotted upon reloading the workspace. Selecting the *Save Workspace* option from the *File* menu calls the native operating system save window where the user is able to save the file in whatever location they wish. The *Load Workspace* function then calls the native load window where the saved file may be located.

5.4.16 Export to PDF

The user of the MDS-GUI is offered numerous methods for retrieving results produced for external purposes. Two such methods are copying any plot to the clipboard of the operating system and printing results directly (See Section 5.4.17). A more in depth method is also available to the user, and this is using the functions from the *Export* menu. Selecting to export in this manner will allow the user to produce a pre-designed PDF document that summarises all results of the GUI session in one easy to follow document. The options are to choose any of the five plotting tab areas to export to a PDF document individually or select to have all occupied areas exported to one longer document (empty areas will not be included). The function makes use of the 'Sweave' package (Leisch, 2002) and the latex document programming language. The user will therefore need to know the very basics of latex usage and have some sort of latex software and interface installed on their system. Upon selecting any of the export options, the instructional message shown in Figure 5.45 will be displayed.

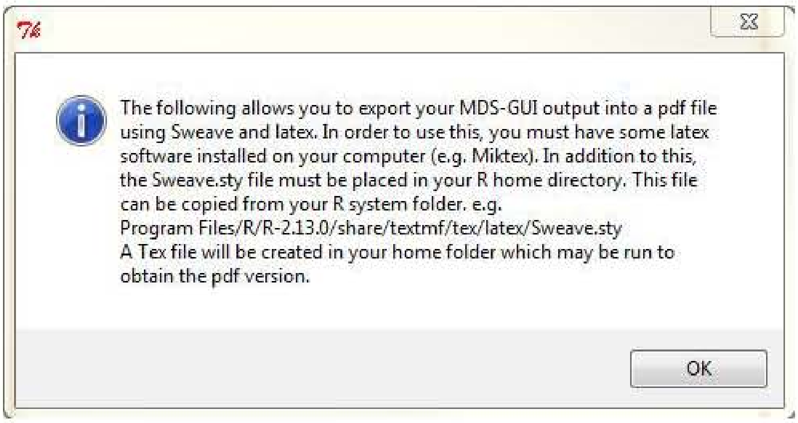
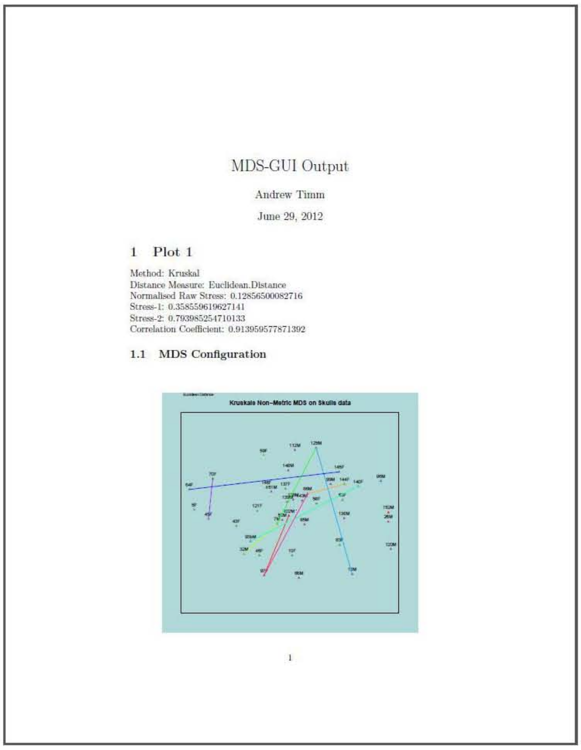


Figure 5.45: Export Instruction Message

The instructions will inform the user that the 'Sweave.sty' file will need to be copied to their home folder from which their current *R*-Environment was called. The current path demonstrated in the example shows "R-2.13.0" which is the environment from which the function was called. If another version of R is used, the path will change accordingly. Following this it instructs that the .tex file which is produced in this home folder should be run through the latex interface. This process should be straight forward to even infrequent latex users. Figures 5.46 and 5.47 demonstrates the layout of the resultant PDF document. General information found in the document include the name of the user of the MDS-GUI instance and the date of the research. Each included plotting area occupies two pages, where the plotting area specific information includes: the method of MDS; the distance metric; all three stress values (Normalised Raw Stress, Stress-1 and Stress-2) and correlation coefficient of the configuration; the MDS configuration; the Shepard Diagram and finally the Scree-Plot.


MDS-GUI Output

Andrew Timm

June 29, 2012

1 Plot 1

Method: Kruskal
Distance Measure: Euclidean.Distance
Normalised Raw Stress: 0.12856500082716
Stress-1: 0.358559619627141
Stress-2: 0.793985254710133
Correlation Coefficient: 0.913959577871392

1.1 MDS Configuration

1

Figure 5.46: PDF Output: Plot1-Page1

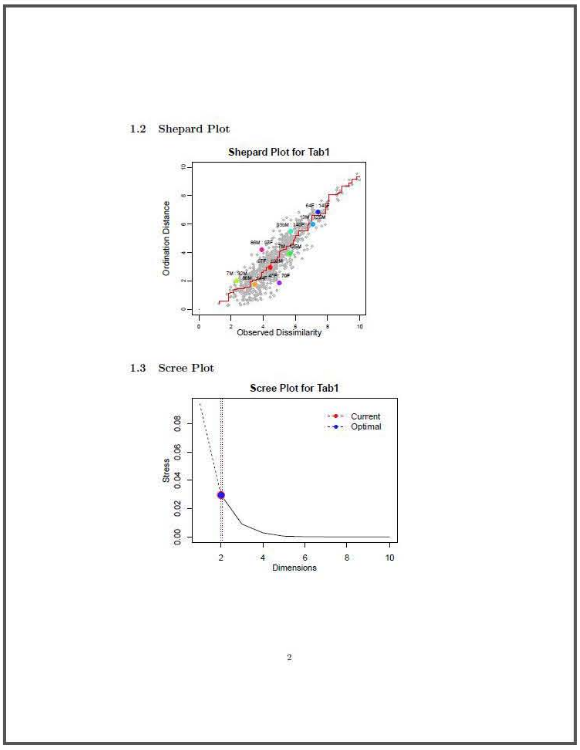


Figure 5.47: PDF Output: Plot1-Page2

5.4.17 Print

The Print function, found in the *File* menu, works very simply. It is designed to allow the user to quickly obtain a copy of the active configuration of the MDS-GUI. Selecting *Print* from the menu will call the standard printing options window native to whichever operating system is in use. The user, as is standard, is prompted to select which printer is preferred from a list of those available to the machine. Other options available from this menu will include number of copies, layout of the page and so on. As the function is designed to be quick the configuration of the focused plotting tab is printed as is, without the inclusion of any other plots or statistics. The default dimensions of the plot takes up the entire width of the allowable margins of an A4 page, and has a proportional height. Any user wanting more detailed printable output is urged to rather make use of the *Export* options described in Section 5.4.16.

5.4.18 Supporting Plotting Features

Each plot in the MDS-GUI, both in the main plotting area and secondary tabs, have a few supporting features that add to the usability of the software. The first is the ability to copy the plot to the clipboard of the operating system. This feature is accessed through the right click menu of the plot and allows the plot to be exported into any external program, such as Microsoft Word.

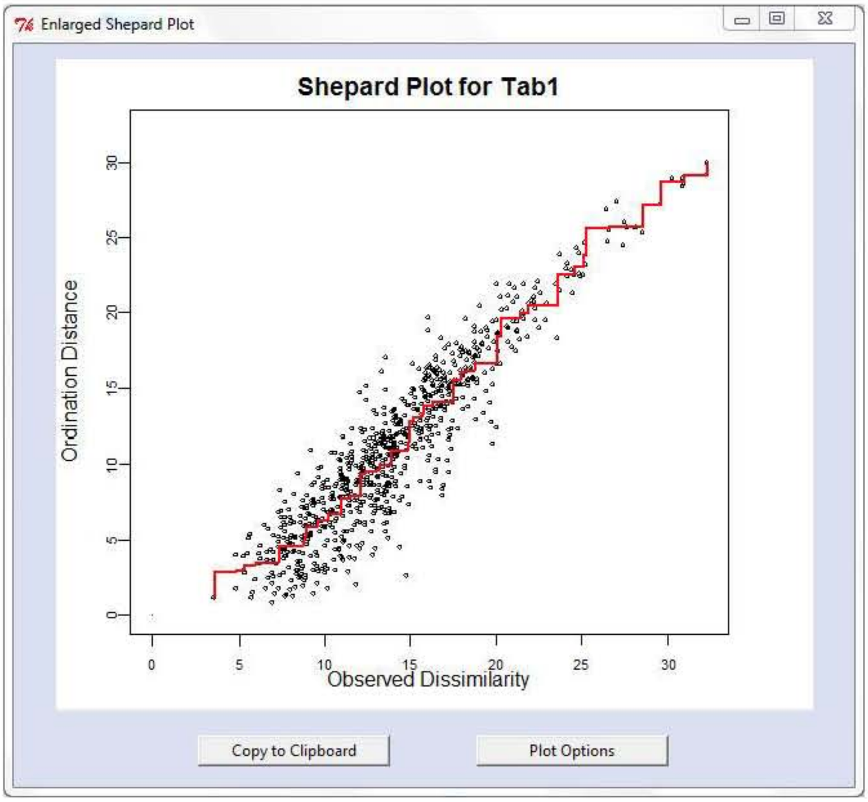


Figure 5.48: Popped-Out Plot

Each plot may also be popped-out from the main GUI itself forming a separate window. This option is also accessed via each right click menu. The advantage is mainly aimed at the individual plots found in the Secondary Plotting Area which all relate to the active main plotting tab. Usually only one of these may be observed at a time however, when they have been popped out, the user may observe many plots simultaneously. All popped-out window may be resized with the aspect ratio of the plot preserved. In addition, where applicable, the popped out plot will show the iterative changes as would have been shown in the MDS-GUI built-in version. Figure 5.48 gives the example of the Shepard Plot after being popped-out from the GUI.

5.5 Coding

Chapter 4 speaks extensively about the various computer coding languages and packages used to create the MDS-GUI and the **MDSGUI** *R* package. This short Section however will discuss some of the aspects around the actual coding of the MDS-GUI itself.

5.5.1 MDS-GUI Development

The final product of the MDS-GUI has the entire extent of the software called by the single function `MDSGUI`. During development however, scripts and *R*-Workspaces were used throughout the time consuming process. Developing such a GUI required two separate components to the code. The first was the group of functions that control all process performed by the software. The number of relevant functions amounted to roughly 250, many of which had further sub-functions built within them. These functions were stored within an *R*-Workspace. The second component of development was the script of code that built the front-end of the MDS-GUI. This script may have only been run once all functions were loaded, as each element of the GUI was built, it had the appropriate functions associated with it.

In total the functions component amounted to roughly 14000 lines of code and the front-end script 2000 lines. Even modern personal computers require a fair amount of time to process all this code. The *RGui* interface first requires the code be cached into the computer memory before running it through the console. With a relatively short amount of code this caching time is barely noticeable by the user. However, when 14000 lines need to be cached the process takes between 2-5 minutes, depending on the machine. After this, running through the console takes a further 2-5 minutes. This lengthy procedure necessitated the use of *R*-Workspaces, as each function need only be uploaded once. The front-end script on the other hand could not be stored in a workspace and required an individual run through each time. This usually took about thirty seconds.

5.5.2 Challenges

The development of the MDS-GUI was not without its coding based challenges. The most prominent of which was in the adaptation of already existing code. The purpose of this Masters project was specifically in the development of a GUI and not strictly in the development of Multidimensional Scaling performing functions. This allowed the use of pre-existing MDS functions for *R* to be incorporated into the code. The challenge with this, however, was related to the MDS-GUI having features that rely heavily on obtaining information after every iteration of the MDS procedures and the fact that most of the MDS functions for *R* providing information only on the final configuration. This therefore required the adaptation of, usually complicated, code such that it was usable and applicable to the MDS-GUI.

The greatest problem lay in the cases of the functions such as `sammon` and `isoMDS` which called upon *C* code during the execution of the function.

5.6 The MDS-GUI: Version 2

The MDS-GUI, like most software, will always be in need of maintenance and upgrades. The GUI, as it stands, is only *Version 0.1* and continued work on the product will see the development of additional features and major changes.

The first obvious planned additions to the software is in increasing the range of Multidimensional Scaling methods. The first method that will be incorporated will be INDSCAL, as mentioned in Section 3.6. The main obstacle in including this method is in the fact that the input data is required to be in the form of multiple $\mathbf{Z}$ matrices, and not a single one as with the current methods. Allowances for this requirement must therefore be made. Other methods intended for later versions will include Gifi, ALSCAL (Alternating Least Squares Scaling) and Unfolding.

The next MDS related addition will be in including alternative transformation options for the Non-Metric MDS procedures. As it stands, the default and only transformation is an isotonic regression, as represented by the step-function line on the Non-Metric Shepard Diagram. Alternative options will include spline based transformations.

The intention of taking the MDS-GUI to further levels of development is coupled with the understanding that this feat is likely to require a restructuring of the code behind the GUI. As it stands, the GUI is governed by global variables, which become messy and clutter up a workspace. Future developments will require a class system be created which will use methods and objects specific to the MDS-GUI. Although far more complicated, this will improve the performance and reliability of the software.

Currently the MDS-GUI does not make provision for missing values. Missing values can be accounted for in any of the MDS techniques based on minimising some form of stress by incorporating weights. As an example, the general stress function of equation (2.1) becomes

$$\sqrt{\frac{\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} w_{ij}(f(\delta_{ij}) - d_{ij})^2}{scale}}$$

where $w_{ij} = \begin{cases} 1, & \text{if } \delta_{ij} \text{ is not missing.} \\ 0, & \text{if } \delta_{ij} \text{ is missing.} \end{cases}$

Further extensions are possible for other values of w_{ij} as long as $w_{ij} \geq 0$. Future versions of the MDS-GUI will incorporate this technique into all available methods of MDS.

5.7 Similar 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).

5.7.1 iMDS

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 data sets 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/.

5.7.2 XGVis and GGVis

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.

5.8 The MDSGUI Package and Supporting Documentation

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 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 occurred. As a result, the current version of MDS-GUI is only compatible with the 32 bit version of the RGui (R Development Core Team, 2012), 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.

A requirement of any piece of software that is to be made available to the public is that it come with sufficient supporting documentation. This documentation is not only required for referencing and cataloging purposes, but it is also intended to provide any new user to the software with sufficient information to make handling of the product as straight forward as possible. This Section briefly describes the documentation that will be included with the **MDSGUI** package, including those required by CRAN when submitting *R* packages to its database. All supporting documentations can be found in Appendix B.

5.8.1 User Manual

The User Manual that accompanies this package can be accessed directly by the MDS-GUI from the *Help* menu (as well as within the downloaded package folder, but this may be difficult to locate). This manual is intended to provide the user with all information required to use the MDS-GUI confidently. The manual is basically a summarised version of this Chapter as it provides information regarding the menus and features of the software. The document is strictly focused on use of the GUI itself and not on statistical interpretation.

5.8.2 MDSGUI Package Reference Manual

The 'Reference Manual' of *R* packages refers to the documentation for the package that is available from the CRAN website. This document serves to provide all developer information of the package as well as describe all components found within the package, including functions and relevant sets of data. Some packages with numerous functions are found to have lengthy Reference Manual documents, however since the **MDSGUI** contains only the `MDSGUI` function, its accompanying Reference Manual is brief.

5.8.3 Vignette

Some *R* packages are accompanied by a Vignette, which serves to provide further information regarding the use of the software. These documents often are focused more on interpretation of results and include example results from included data.

Chapter 6

Application of the MDS-GUI

Chapter 5 discussed the MDS-GUI and described its structure and various features. This Chapter will aim to demonstrate these capabilities by using the program to analyse different data. Three different data sets have been chosen for analysis, each different in structure and specifically selected to highlight the various aspects of the MDS-GUI output. Full descriptions of each data set and its variables are provided in Appendix A. This Section will report only on the details of the output directly relevant to the data and the MDS results. Information such as 'Time of Process', which is dependent on the machine being used, will not be examined.

6.1 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), Carrol and Wish (2002), Maechler (2009) and Everett (2001). 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 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 asymmetric and symmetric data sets are provided as Tables A4 and A5 of the Appendix, respectively.

6.1.1 Morse-Code: General Analysis

Analysis of the Morse-Code data using the MDS-GUI first requires the data be uploaded into the program. Most regular *R*-users will be aware of the data structure requirements of *R* when uploading data, and these standards also apply to the MDS-GUI. Seeing that the data already comes in the form of the $n \times n$ similarity matrix, the data is loaded through the *Load Similarity Matrix* command in the *Data* menu. The user is then prompted to name their data appropriately. As the data is already in the form of the proximity matrix, all features of the program relating to variables of the data are automatically deactivated and therefore unavailable to the user for the analysis of the Morse-Code data. This includes adjusting the distance calculation method and displaying variable axes.

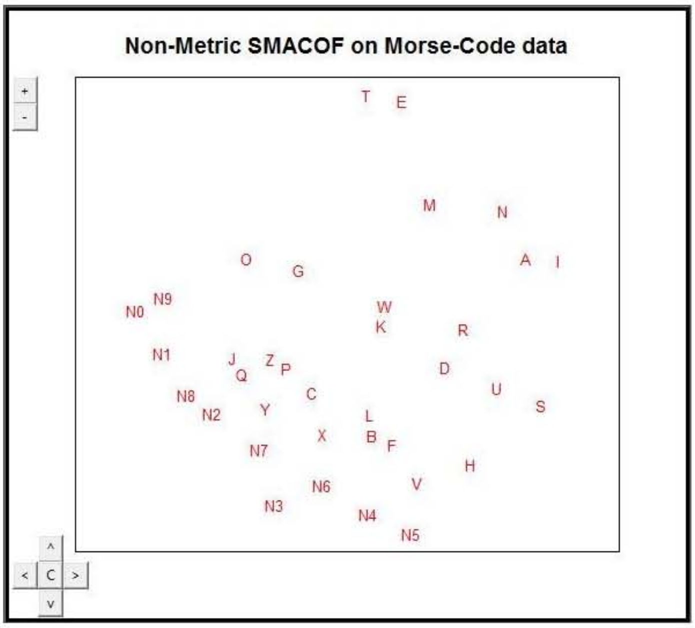
Table 6.1: MDS Stress Values on Morse Code Data

Method	Normalised Raw Stress
Classical Scaling	0.741
Metric SMACOF	0.099
Met. Least Squares Scaling	0.647
Non-Metric SMACOF	0.039
Kruskal's Analysis	0.040
Sammon Mapping	0.090

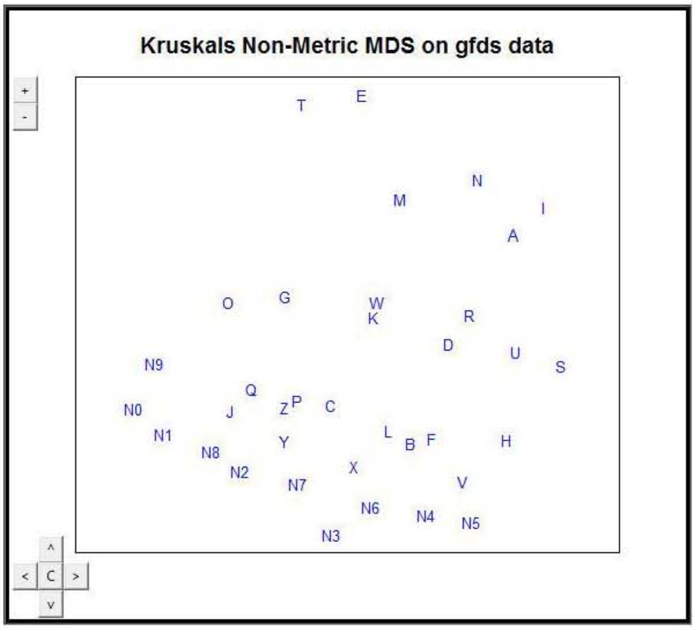
A starting point of many analyses using Multidimensional Scaling is reviewing the results of multiple MDS methods in order to select the most appropriate. The MDS-GUI allows for this process to be performed swiftly and simply, as with multiple plotting areas, the user is able to compare stress values, in different formats, between the various results directly. Table 6.1 shows the Normalised Raw Stress (NRS) values achieved by the various methods for $p = 2$. Inspection of the table reveals that, two of the Non-Metric methods, being Non-Metric SMACOF and Kruskal's Analysis produce the best results, with NRS values of 0.039 and 0.04 . These configurations are displayed in Figure 6.1. Figure 6.2 then goes on to show the result of a Procrustes analysis between the two configurations. Such an analysis is often advisable even when stress values are so similar, as the same stress result may have been achieved through completely different distortions of the proximity information. What is clearly visible from the comparison, however, is that the difference between the two configurations is negligible in this scenario. Due to the fact that the Non-Metric SMACOF result produced the slightly smaller stress value, it will be used for the remainder of the Section.

The Shepard Diagram and Scree Plot for the Non-Metric SMACOF result are provided in Figure 6.3(a) and Figure 6.3(b) respectively. What is immediately noticed when inspecting the Shepard Plot is the shape of the function line of the plot, and the points around it. The first portion of the plot, Observed Distance between 0 and 0.6, sees a relatively flat fitted curve, suggesting that these point pairings have been portrayed shorter than the observed distances. The second portion, Observed Distance between 0.6 and 1, sees a rapid increase in curve gradient, indicating that all point pairings here have been overstated by the MDS procedure. It is important to note that this shape of Shepard Plot is common to all other results coming from the alternative MDS methods on the *Morse-Code Data*. The Non-Metric SMACOF result, however, displays the tightest grouping of the points around this function line. As the stress formula for non-metric methods comprises a distance component of the squared and summed vertical distance between point and function line, the small stress value of the Non-Metric SMACOF is easily explained. The poorer results of the metric methods is also understood, as the stress calculations involve the direct comparison between Observed Dissimilarity and Ordination distance.

The Scree Plot shows no surprising results and reveals no obvious 'kinks' on the curve. Analysis of the curve angle at each dimension suggests that



(a) Non-Metric SMACOF Result



(b) Kruskal's Analysis Result

Figure 6.1: Morse-Code Data: Best Results

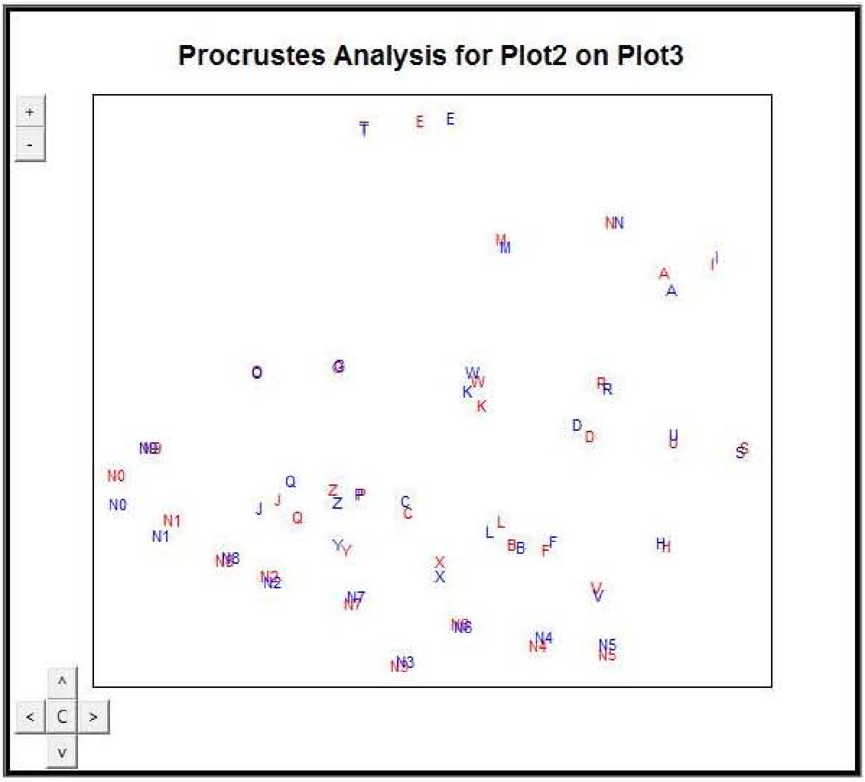


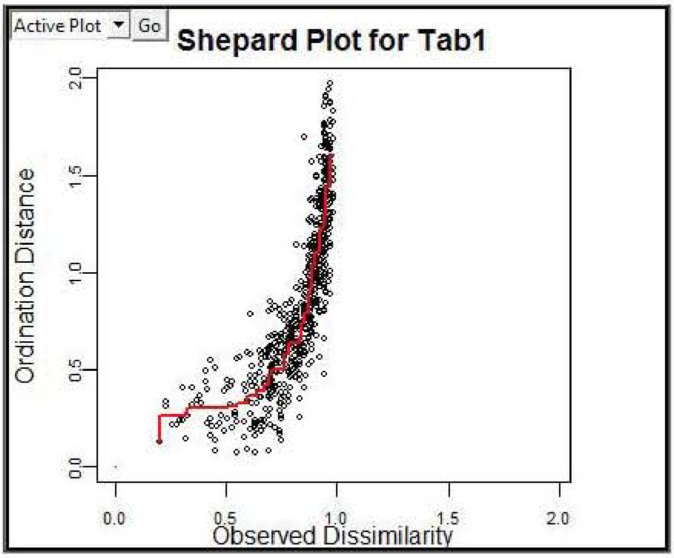
Figure 6.2: Morse Code: Procrustes Analysis

the optimum dimensions for the data is two. No further analysis into output with higher dimensions is necessary.

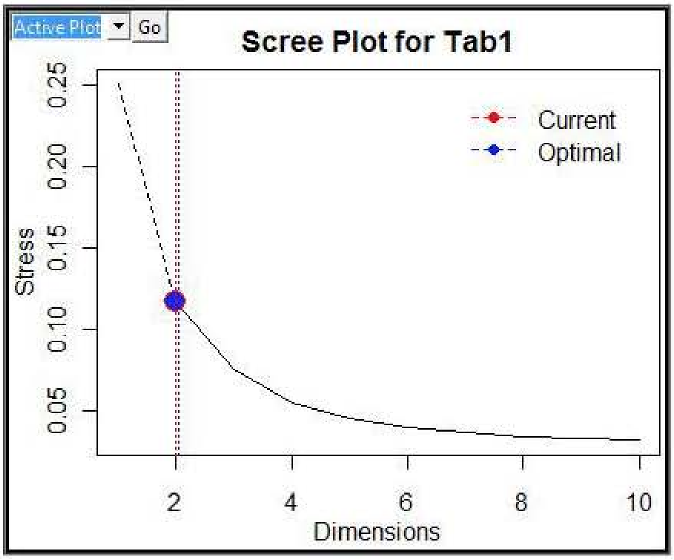
6.1.2 Morse-Code: Configuration Analysis

Inspection of the configuration (Figure 6.1(a)) without any particular emphasis reveals only a few basic observations. The numbers are clearly distinguishable as they form a line on the outskirts of the configuration. The letters T and E are also observed to be more removed from the main clustering than any other points. It is however clear that some form of classification of the points is necessary to aid the interpretation process.

The first attempted classification is in defining the categories of the points as the type of symbol that they are. Specifically, the three categories are 'Number', 'Consonant' and 'Vowel'. The data file will then include an additional column representing the Categories of the data. The data is up-



(a) Non-Metric SMACOF: Shepard Plot



(b) Non-Metric SMACOF: Scree Plot

Figure 6.3: Morse-Code Data: Diagnostic Plots

loaded once more into the MDS-GUI and the box is checked, in the *New Active Similarity Matrix Options* window, indicating that a column of categories is present. Each point is then automatically colour coded according to its respective category. The updated Non-Metric SMACOF result is shown in Figure 6.4.

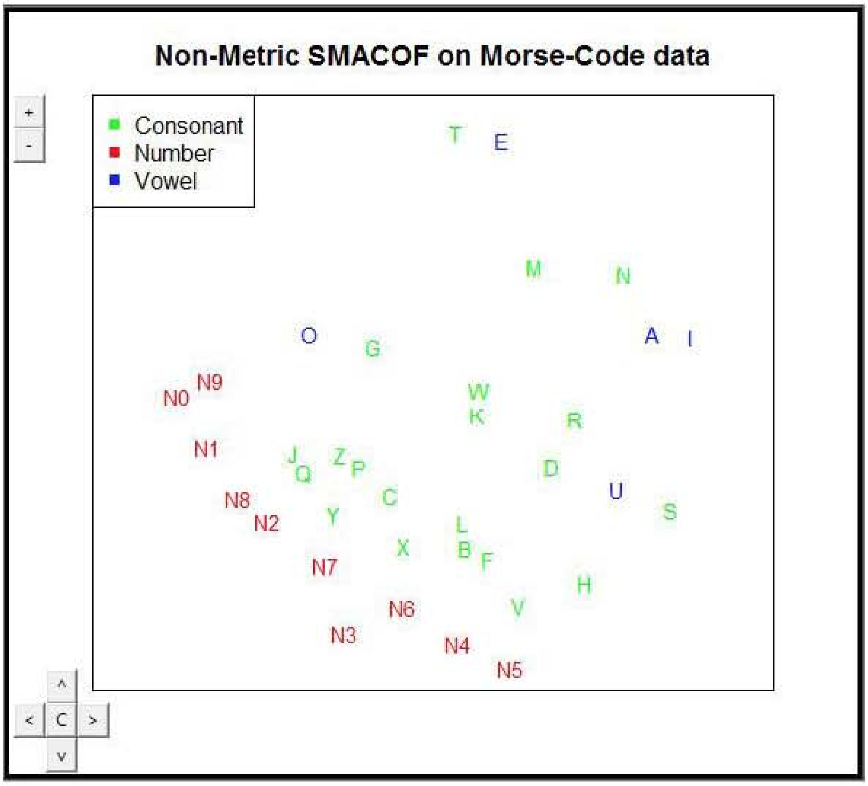


Figure 6.4: Morse Code: Symbol Categories

The categorised version provides a slightly better idea of the composition of the configuration. The numbers are again clearly defined and interesting the majority of the consonants are grouped tightly towards the center of the configuration. The vowels are not clearly grouped together, but it is noticed that each vowel is located on the outskirts of the configuration, indicating that in general they were perceived as non-similar to the majority of the other points. A more detailed category system is then undertaken in order to identify a more profound interpretation and understanding of the MDS result. In order to do this, a more detailed look is required at the composition of the Morse code symbols themselves. Table A.3 of the appendix provides the codes for each of the 36 symbols. It is noticed that each symbol is represented by a series of between 1 and 5 dashes and/or dots. A reasonable hypothesis is that, when determining the similarity between two simultaneous sequences, the length of the two will be significant to the result. Each point of the data set is now provided with a category indicating whether its Morse-Code sequence is comprised of 1,2,3,4 or 5 elements. The reloaded data in the MDS-GUI produces Figure 6.5. The background colour of the *Main Plotting Area* of the MDS-GUI is adjusted to a gray so as to emphasize the difference in colour of points.

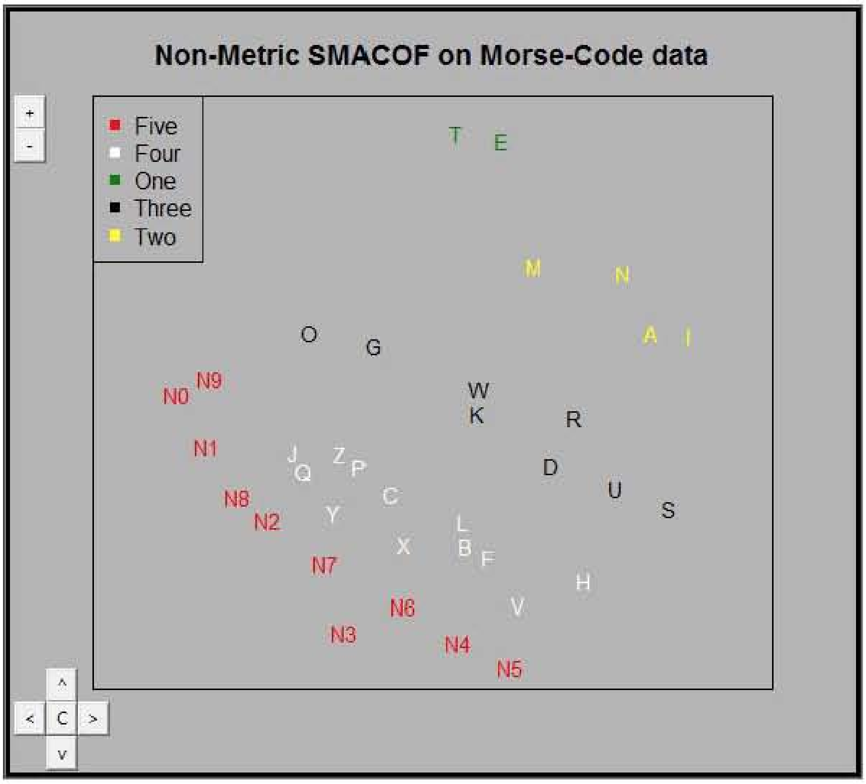
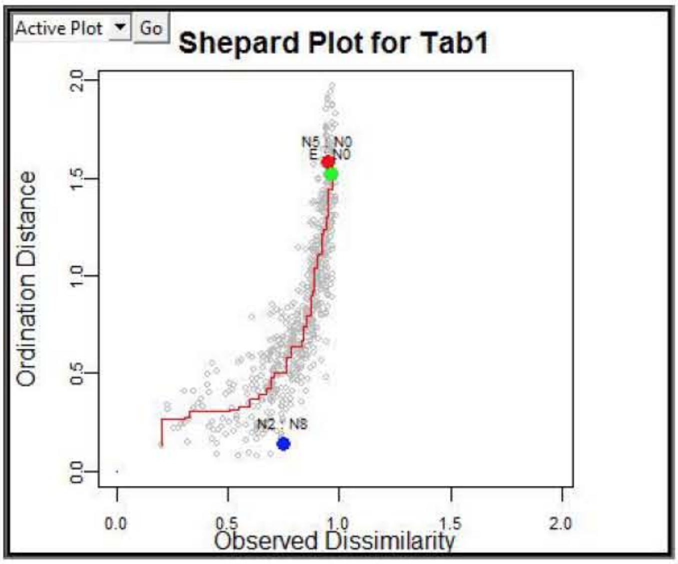
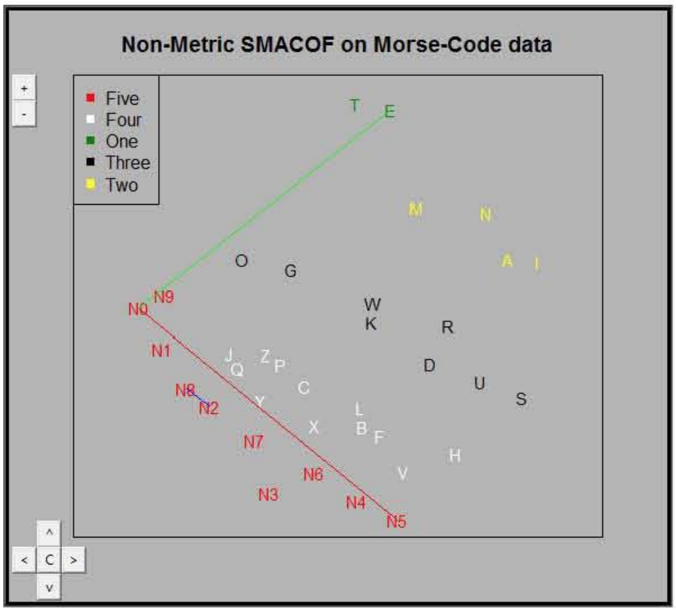


Figure 6.5: Morse Code: Length Categories

This result produces a far better depiction of how the respondents of the investigation responded. The configuration is very clearly influenced by the length of the code, with each length category grouped together. To be specific: sequences with five elements are red; with four elements are white; with three elements are black; with two elements are yellow; and with one element is green. We therefore see that not only are the categories grouped, but the groups are arranged such that they are ordered. That is, the single element group is furthest from the five element group, and so on.



(a) Morse-Code: Shepard Diagram with Labeled Points



(b) Morse-Code: Configuration with Labeled Points

Figure 6.6: Morse-Code Data: Labeled Points

Some point pairings require detailed analysis. Figure 6.6 shows the labeling of two points on the Shepard Diagram, with the corresponding distance illustrated on the configuration mapping. These points were added by selecting the *Label Specific Point* option of the Shepard Plot right click menu. The first pairing is that of *N0*:*N5*, which has the Shepard point and line coloured 'red'. Each of the ten number sequences are comprised of five elements (letters contain up to four elements). Of these ten sequences, one would expect the *N0*:*N5* combination to be perceived as the least similar numbers, as the *N5* sequence consists of five dots and *N0* consists of five dashes. This prediction is confirmed as the two points are the furthest apart of all the numbers. All other number combinations consist of different combinations of dots and dashes and are therefore more easily confused. In particular, the *N2* sequence consists of two dots and three dashes, while the *N8* sequence is three dashes and two dots. These two are often confused, and therefore are the closest of the number pairings. This pairing is coloured blue on the Shepard Plot and configuration. Finally, the *N0*:*E* combination should be of interest as one would expect it to be the two least similar objects of the data. *N0*, as described, consists of five dashes, while *E* has only one single dot. The result (colour coded green) however reveals that the MDS procedure did not capture the extent of this distance. The Shepard Plot reveals this point to indeed have one of the highest observed distances, yet is far from the highest ordination distance. The interesting point here is that while the ordination distance of the point is not the greatest of all the ordination distances, it is still far higher than its specific observed distance, meaning that some lesser observed distance pairings have been exaggerated even more drastically than *N0*:*E*. This further confirms the conclusions drawn about the Shepard Plot in that the majority of points have been severely overstated by the Non-Metric SMACOF procedure.

6.2 SynTReN Microarray Data

A form of technology that has undergone great improvements over the last few years is that of microarrays. These experiments produce vast amounts of gene expression data which have given cause for the involvement of statisticians using multivariate analysis to interpret the results. The data from microarray experiments involves a set of genes, and the level in which they are 'expressed' over a set of samples. Gene Association Networks may then be produced indicating the relationships between the various genes (relationships are in terms of 'activation', 'repression', etc.). Therefore, the genes are the subjects of the data and the samples act as the variables. Interested readers are referred to any of the numerous texts available on microarray data and gene association networks. Two such examples are the works of Allison et al. (2006) and Causton et al. (2003).

The SynTReN (Synthetic Transciptional Regulatory Networks) computer program (Van den Bulcke et al., 2006) is popular amongst bioinformaticians as it is used to simulate microarray-like data. The advantage of generating data is that statistical methods for determining significant gene interactions may be tested on the data in order to test each method's capabilities. This is possible due to the fact that the simulated data is based on a true 'Source Network' and the 'true' structure of the network is thus already known. The data used in this Section is an example of the data generated by SynTReN. The data set was set to have fifty genes and one hundred samples from the Ecoli Source Network with default settings for all noise parameters. The resulting matrix was thus 50×100. The true structure of the network is shown in Figure 6.7. Lines between nodes of the network indicate specific interactions, but the relevant feature in this context is the groupings of the genes. Three primary groupings of genes occur in this example, each with a central gene. In addition two connecting genes exist. Multidimensional Scaling procedures will therefore be considered successful when the groupings have been clearly distinguished in the resulting configuration. Simple colour coding has been used for easy differentiation between groupings.

The 'variables' of microarray data represent the people from which a sample was taken. This means that attempting to plot variable axes would show the regression line of each person through the configuration. This information is irrelevant in such an experiment as gene interaction is the only information important to the experimenter. Interpretation of variables will therefore not come into the analysis of this data, but rather the interesting graphical configuration output will be under focus.

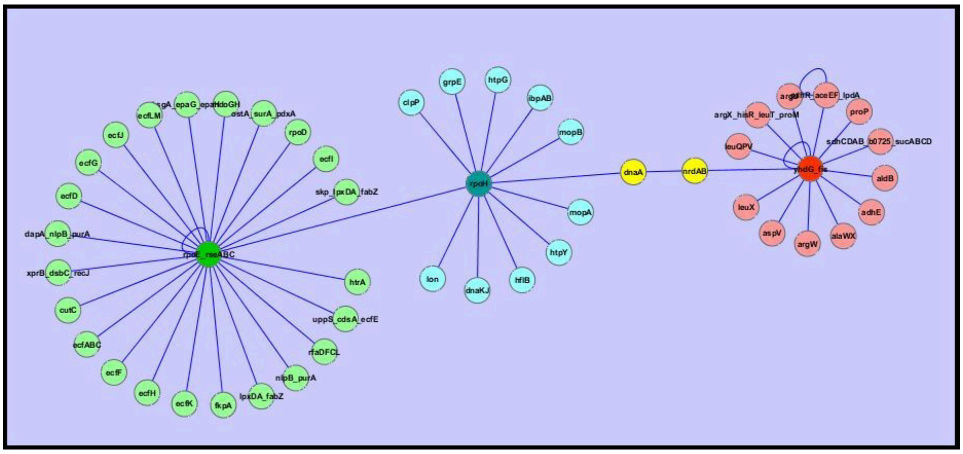


Figure 6.7: True Gene Association Network

6.2.1 SynTReN Microarray: General Analysis

The structure of microarray data is asymmetric and non-square, meaning that uploading the data set into the MDS-GUI requires use of the *Load Dataset* option in the *Data* menu. Once loaded, the $n \times n$ dissimilarity matrix, $\mathbf{\Delta}$, is calculated automatically using whichever calculation measure is under current selection (Euclidean Distance by default). Since this data requires the dissimilarity calculation, a comparison of different MDS methods and different proximity calculations can be performed. Table 6.2 gives the Normalised Raw Stress (NRS) values, when $p = 2$, for each of the Metric and Non-Metric MDS methods when performed using a proximity matrix derived using a selection of four dissimilarity calculation measures (Euclidean Distance, City-Block Metric, Canberra Metric and Wave-Hedges). In all non-Classical Scaling cases, the starting configuration supplied is the Classical Scaling result.

Table 6.2: NRS Values on Microarray Data

Method	Euclidean	City-Block	Canberra	Wave-Hedges
Classical Scaling	0.055	0.040	0.119	0.201
Metric SMACOF	0.014	0.011	0.018	0.032
Met. Least Squares Scaling	0.045	0.029	0.097	0.170
Non-Metric SMACOF	0.004	0.004	0.004	0.005
Kruskal's Analysis	0.004	0.004	0.004	0.005
Sammon Mapping	0.011	0.007	0.014	0.026

The table reveals some interesting observations. Firstly, and unsurprisingly, in each case the Stress value for Classical Scaling result is higher than all other methods. This is due to the fact that by definition, each iteration of the MDS procedure will produce a stress value less than or equal to the previous iteration. This means that since the configuration obtained from Classical Scaling is used as iteration 0, the result may only be improved upon. The two results that appear to be most successful, in each dissimilarity measure case, are those obtained from Kruskal's Analysis and Non-Metric SMACOF. The Stress values from both configurations are 0.004 for the Euclidean, City-Block and Canberra cases, and 0.005 when Wave-Hedges is used. According to the guidelines of stress interpretation, these results are considered very good.

Observing the figures associated with Kruskal's Analysis, an identical Stress value (to the third decimal place) is obtained when using three different dissimilarity measures to calculate the proximity matrix. One might expect an explanation of this to be that Kruskal's Analysis produces near identical results regardless of the difference in the input proximity data. Figure 6.8 however, shows a Procrustes Analysis of the configuration produced by Kruskal's Analysis with a proximity matrix calculated using the City-Block Metric (red) and Kruskal's Analysis with a proximity matrix calculated using the Canberra Metric (blue). The analysis reveals that the two configurations, while exhibiting similar structures, are notably different from one another. Scenarios such as these demonstrate the tendency of Multidimensional Scaling to be subject to finding local minima during optimisation, regardless of how small the resulting stress value is. The tolerance restricting each of the methods in the analysis was the default for the MDS-GUI, being 0.00001. An interested party may be inclined to adjust this tolerance to a far smaller value in the hopes of observing observations closer to a global minimum, however due to the acceptable stress value of these configurations in their current state, this is not necessary.

The NRS value achieved by Metric SMACOF using the City-Block metric, however, is also exceptionally low at 0.011. Since Non-Metric MDS only requires the rank order of distances in the configuration to match given dissimilarities, the stress values will always be smaller for Non-Metric versus

Metric MDS. In the remainder of this example a metric method of MDS is used on the SynTReN microarray data. The result of Metric SMACOF with the City-Block metric will therefore be utilised despite not technically producing the lowest stress value. In practice, Metric MDS methods are only utilised when specifically required by the researcher, as is the case here.

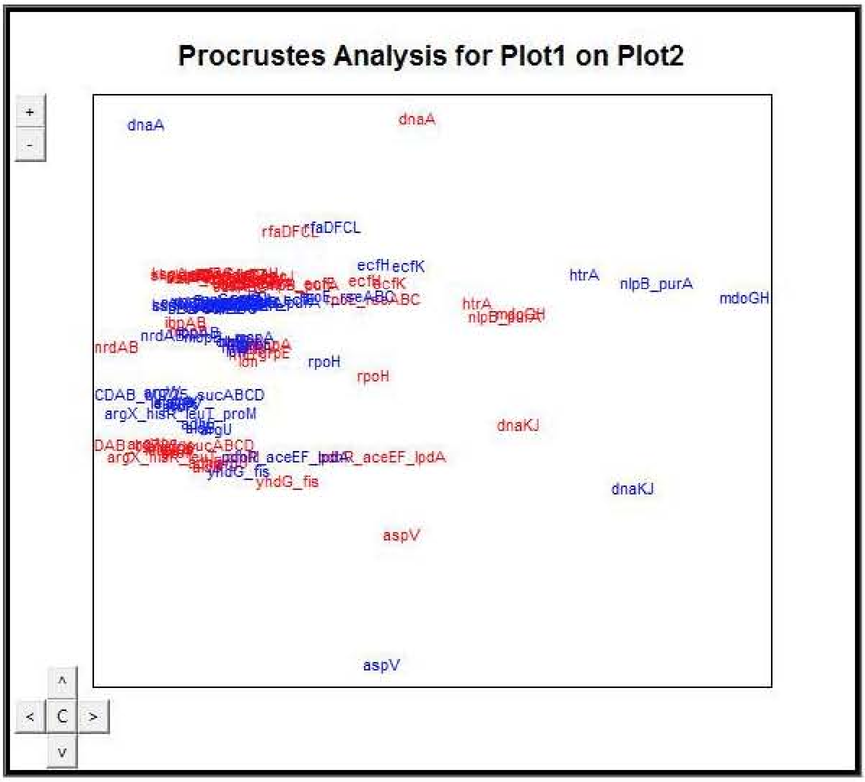


Figure 6.8: MicArray: Procrustes Analysis

The configuration in Figure 6.9 does not easily highlight any obvious patterns. Small groupings of points are present, yet their meaning is not necessarily obvious without categorical classification. The Shepard Plot for the Configuration (with proximities measured using the City-Block Metric) and the Scree Plot are shown in Figures 6.10(a) and 6.10(b) respectively. The Shepard Plot shows the vast majority of point pairings lying very close to the identity function line, accounting for the extremely small Stress Value. A small number of points deviate from this trending function curve and will require further analysis. The Scree Curve exhibits a very definite 'kink' at

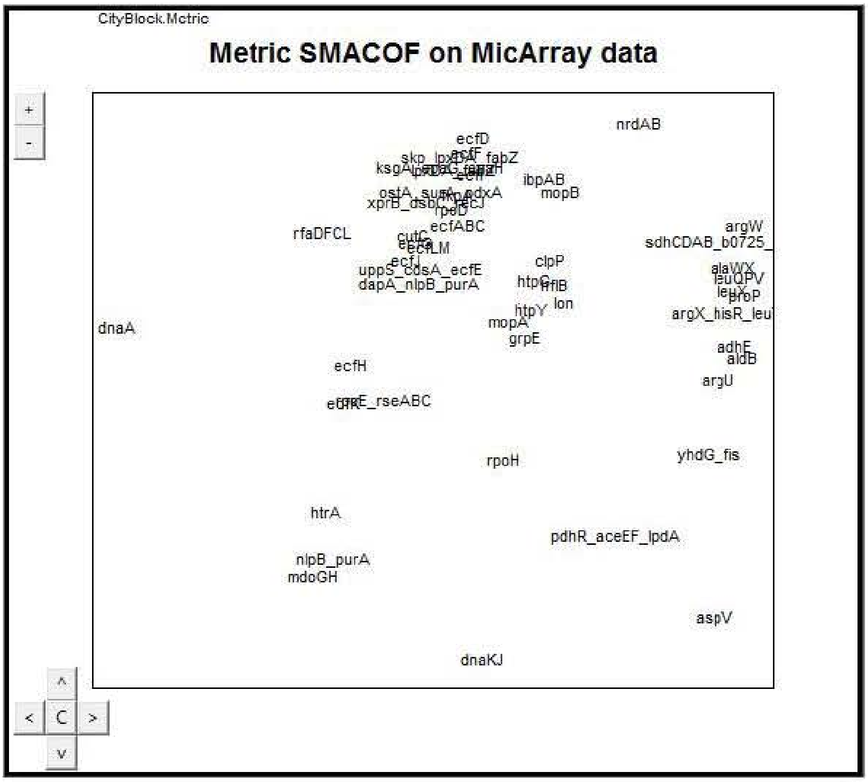
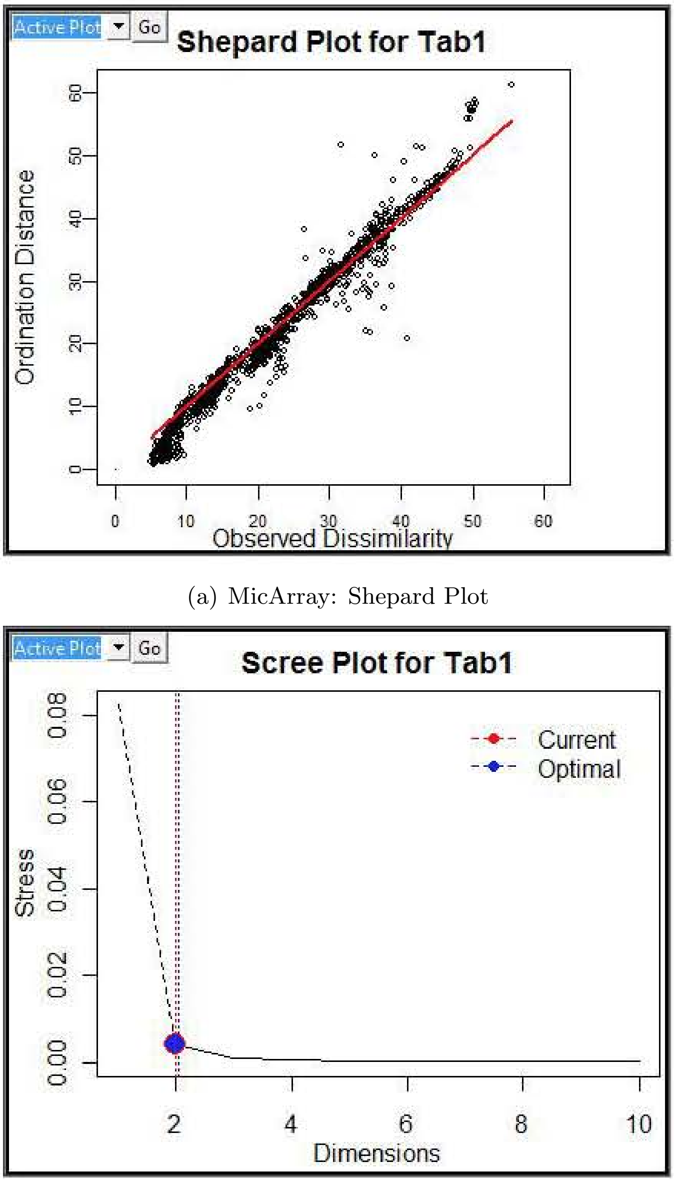


Figure 6.9: MicArray: Metric SMACOF (City-Block Metric)

the second dimension, indicating clearly that $p = 2$ is the optimum setting for this SynTReN Microarray data.

6.2.2 SynTReN Microarray: Configuration Analysis

The nature of the SynTReN Microarray data allows for a convenient way of interpreting any MDS configuration in terms of structure. The presence of the 'True Network' (Figure 6.7) allows the researcher to directly compare the configuration with the truth and make direct comparisons. As described, the MDS procedure can be considered successful in this instance when all structures of the 'True Network' have been distinguished adequately. The use of colour in the MDS-GUI with the same colour coding as in Figure 6.7 allows for the most convenient comparison. A category column is added to the data set indicating which group of the true network each gene (object) belongs to. The categories were termed: 'Left Group', 'Left Group Center',



(a) MicArray: Shepard Plot

(b) MicArray: Scree Plot

Figure 6.10: MicArray: Diagnostic Plots

'Middle Group', 'Middle Group Center', 'Right Group', 'Right Group Center' and 'Connector'. The data is then reloaded into the MDS-GUI with the presence of categories selected. Following this, using the *Category Colours* feature, the user is able to quickly match each category (and background) with the corresponding colour of the ‘True Network’. Figure 6.11 shows the Kruskal’s Analysis result with these changes in effect.

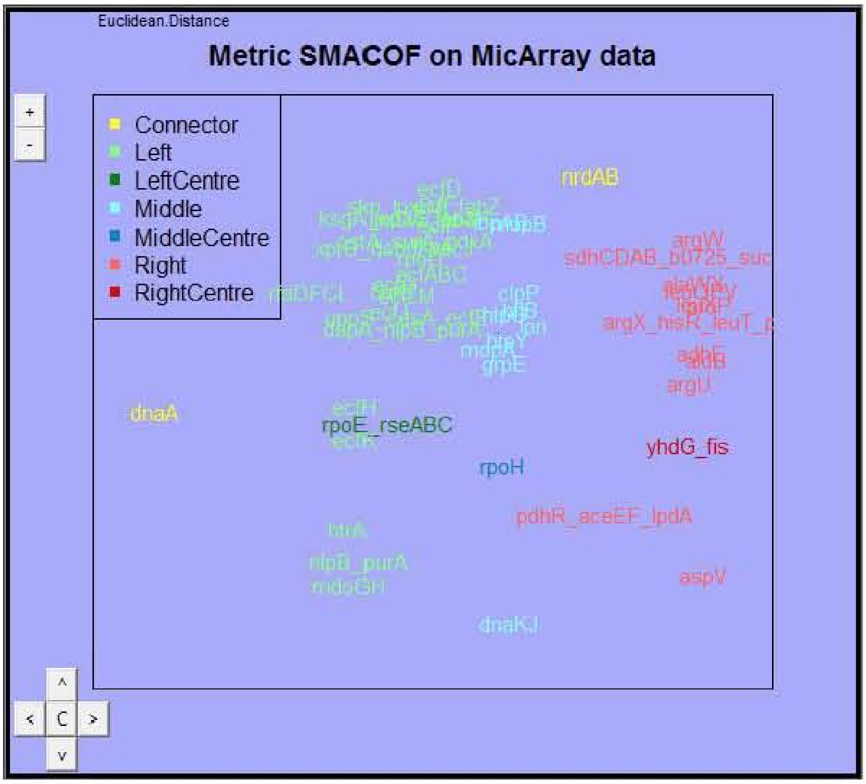
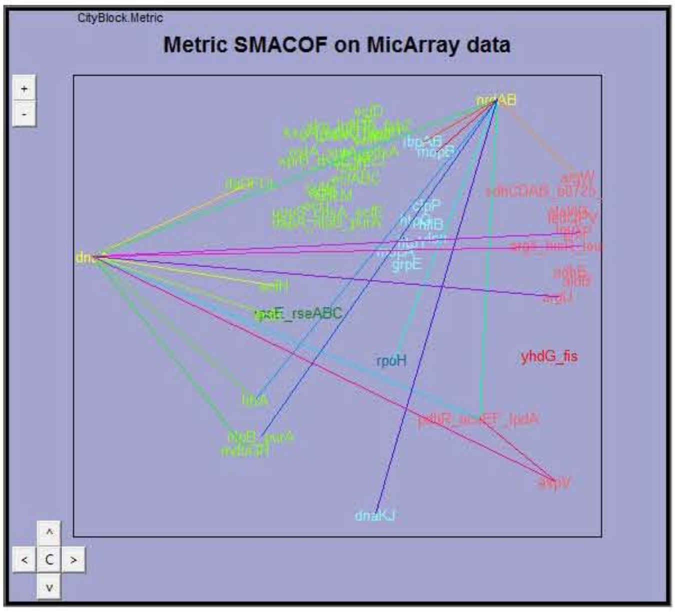


Figure 6.11: MicArray: Metric SMACOF with Coloured Groups

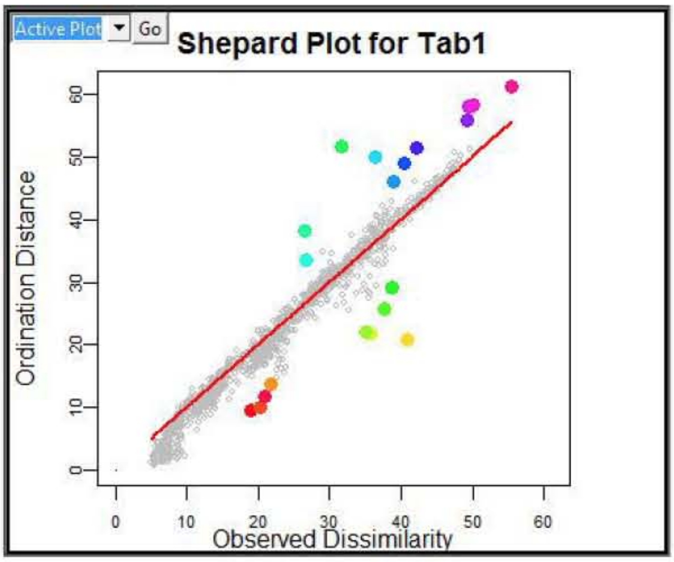
The addition of categories has a profound effect in the interpretation of the configuration. The three major clusters found in Figure 6.7 have clearly been defined in the MDS configuration. For convenience, the configuration is rotated in such a way that the groupings are in the same graphical locations as the true network. This rotation is performed through the *Rotation and Reflection* menu window accessed via the *Main Plot Options* menu of the configuration area. It is clear to see that the groupings are near perfectly separated and are in the same ordering as the true network. In addition it is also seen that the central gene of each grouping is further distinguished in that they are all slightly removed from the majority of their respective clusters. What is clear however is that the location of the two ‘Connector Genes’, which have been coloured ‘yellow’, are notably different from their corresponding positions in Figure 6.7. A valid hypothesis therefore might be that these two genes have been poorly located by the MDS procedure and may be responsible for the majority of Stress present in the configuration. In order to explore this theory, the highlighting points on the Shepard Plot feature is put to use. As already mentioned, the Shepard Plot shows a small number of points that deviate clearly from the grouping surrounding the function line. Each of these deviating points are highlighted (with corresponding lines drawn between points on the configuration). The highlighted Shepard Plot and corresponding configuration plot are both shown in Figure 6.12.

Highlighting these deviating point pairings reveals that almost all cases have a 'Connector Gene' as one of the two involved points. This evidence strongly suggests that these two objects are in fact poorly represented by the MDS process and that their specific locations should not be considered when drawing conclusions of the data in terms of the MDS based configuration. In truth, the difficulty of accounting for these two genes is understandable seeing that structurally, they are not as strongly defined as every other gene in the network.

Interpretation of the Scree Plot in Figure 6.10(b) revealed that no analysis when $p > 2$ be necessary. The three dimensional analysis, shown in Figure 6.13, was performed purely for interests sake. This particular figure is a Windows screenshot of the RGL result, housed in the *RGui*. Manual rotation of the environment allows the user to easily observe that each of the three main clusters have unique locations in the first, second and third dimensions, making each even more differentiable in a three dimensional space. Furthermore, the two 'Connector' genes that were placed inaccurately are now found to be satisfactorily placed (especially in the view shown in Figure 6.13). The Normalised Raw Stress, to three decimal points, is reported to be less than 0.001 according to the MDS-GUI. The Shepard Plot for this configuration is given in Figure 6.14, where it is observed that all points relating to the two 'Connector' genes are no longer deviants from the transformation curve. The drop in stress is thus explained and the configuration can now be seen to be about as accurate as possible.



(a) Deviating Pairs: Configuration



(b) Deviating Pairs: Shepard Diagram

Figure 6.12: MicArray: Deviating Pairs

6.3 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

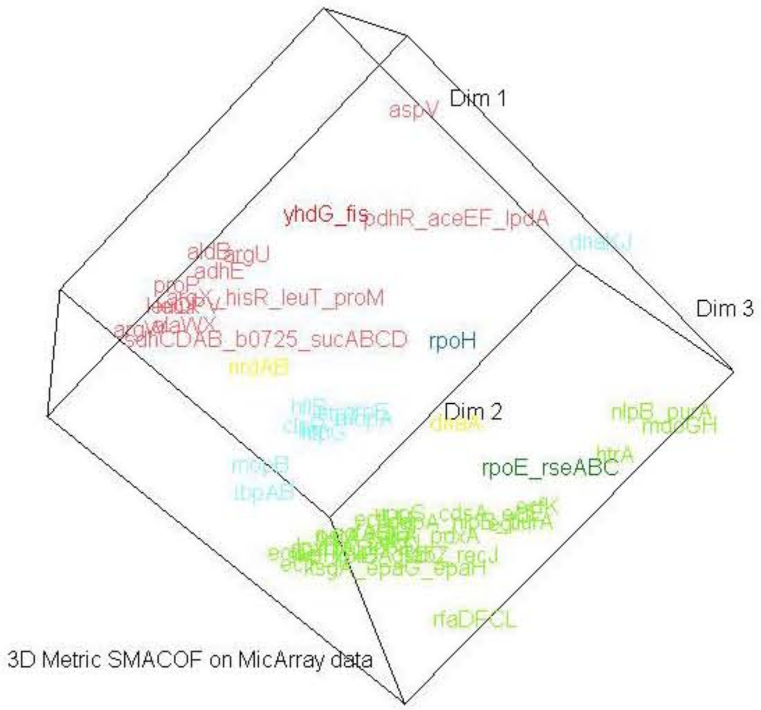


Figure 6.13: MicArray: p=3 Configuration

constitute various nutritional components and are all measured on the ratio scale, as significant zero's exist in each case. The last is an ordered 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. See Tables A.6 and A.7 of the Appendix for details about the object and variable names.

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

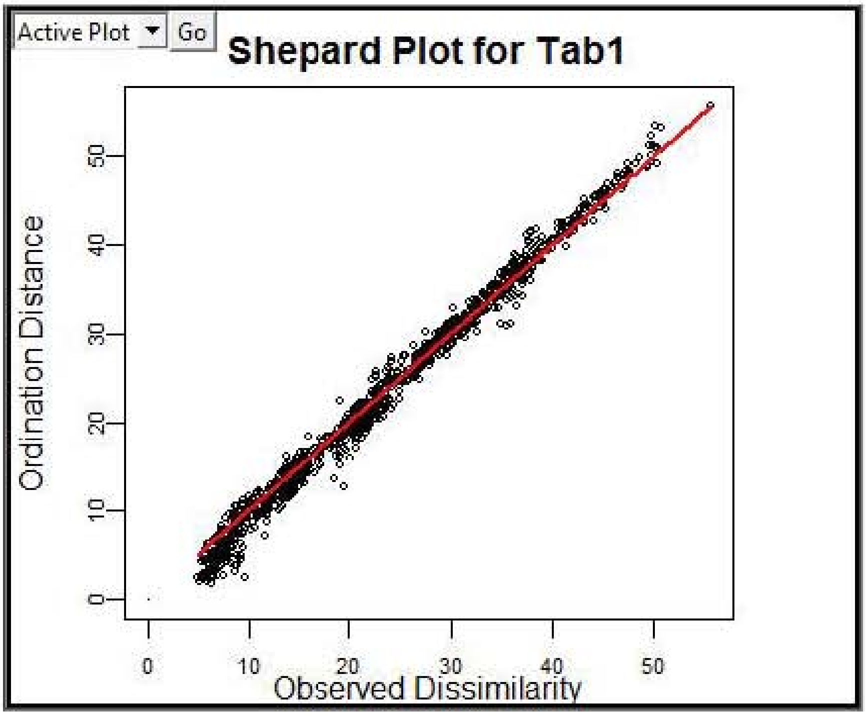


Figure 6.14: MicArray: p=3 Shepard Plot

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.

6.3.1 Cereal Data: General Analysis

The structure of the Cereal Data is similar to the SynTReN Microarray data in that it is in the form of an $n \times m$ $\mathbf{Z}$ matrix. The data is therefore also uploaded into the MDS-GUI with the use of the *Load Dataset* option. The $n \times n$ dissimilarity matrix, $\mathbf{\Delta}$, is then calculated automatically using whichever dissimilarity matrix calculation method has been selected in the *Dissimilarity Matrix Calculation* drop down menu. Once again, the fact that the input dissimilarity matrix is subject to this calculation means that the results of a number of measurement options need to be compared in determining the most appropriate dissimilarity measure for the data. Table 6.3

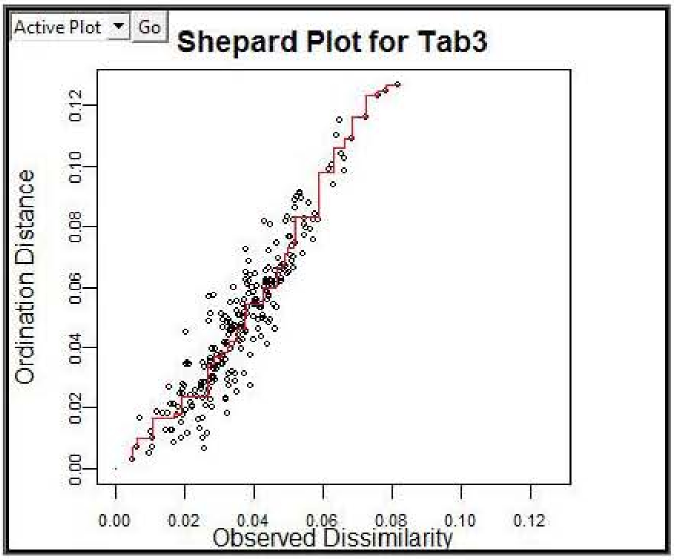
shows the STRESS-2 values for each of the metric and non-metric MDS methods of the MDS-GUI for $p = 2$. STRESS-2 has been selected as the accuracy measurement criteria for this case-study in order to demonstrate the use of a version of STRESS that is notably stricter than the Normalised Raw Stress criteria. The four measurement methods selected for this study are the Euclidean Metric, Angular Separation, Bray-Curtis Distance and Bhattacharyya Distance. As before, the starting configuration for each case is the configuration produced by Classical Scaling using the same proximity matrix input. The tolerance is the default value of 0.00001.

Table 6.3: Stress-2 Values on Cereal Data

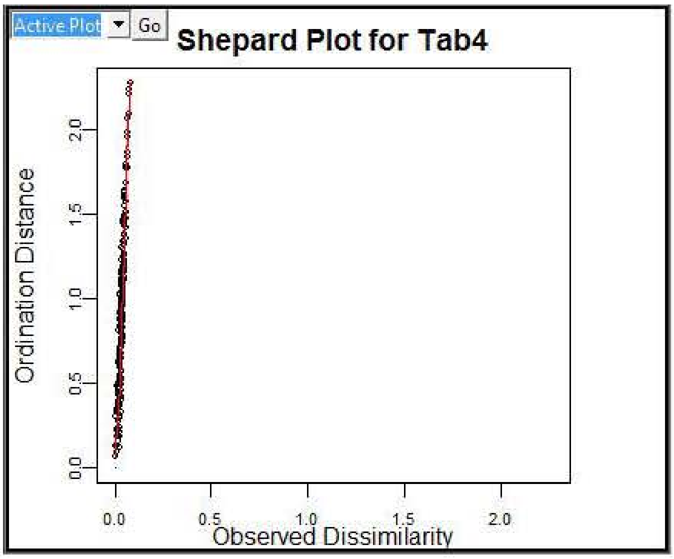
Method	Euclidean	AngularSeparation	Bray-Curtis	Bhattacharyya
Classical Scaling	1.000	0.452	0.617	1.000
Metric SMACOF	0.522	0.390	0.424	0.515
Met. Least Squares Scaling	0.695	0.419	0.459	0.718
Non-Metric SMACOF	0.304	0.332	0.310	0.310
Kruskal's Analysis	0.304	0.332	0.310	0.310
Sammon Mapping	0.414	0.349	0.357	0.412

Inspection of the table suggests that, once again, Kruskal's Analysis and Non-Metric SMACOF produce the best results. Strictly, the dissimilarity matrix calculated with the Euclidean Metric is most appropriate, however in order to maintain a level of variety, the result using the Bray-Curtis measure will be carried through the case study. The difference in STRESS-2 values is a mere 0.007 and so the analysis can be made without any significant loss of accuracy. The Bray-Curtis measure should technically be used exclusively in non-metric studies, which rules out the use of the first three MDS methods. In order to determine which of the Kruskal's Analysis and Non-Metric SMACOF should be used in the study, the respective Shepard Diagrams are observed. These are provided in Figures 6.15(a) and 6.15(b)

The Shepard Diagram for the Kruskal's configuration (Figure 6.15(a)) is reasonably well spread out. The Shepard Plot for the Non-Metric SMACOF, however, has very little visual spread with all the points lying within a small horizontal range. From a Multidimensional Scaling point of view, this has no effect in the accuracy of the model due to the fact that the method is non-metric and therefore it is only the distance between the transformed



(a) Kruskal's Analysis (Bray-Curtis)



(b) Non-Metric SMACOF (Bray-Curtis)

Figure 6.15: Cereal Data: Shepard Plot Comparison

$\hat{d}$ and the d values that are taken into account. However, from an interpretation point of view, this shape of Shepard Plot is not ideal as it makes differentiating between points of the plot more difficult than a case such as that found in Figure 6.15(a). For this reason, the Kruskal's Analysis results are found to have an easier interpretation and will therefore be used for the remainder of this Section. For the sake of interest, it should be noted that if the Shepard Plot in Figure 6.15(b) were produced by a metric method of MDS, the Stress value would be very high and the configuration would be considered poor. This is due to the fact that the distance between every point pairing has been greatly overstated, as the ordination distances are far higher than their corresponding observed distances.

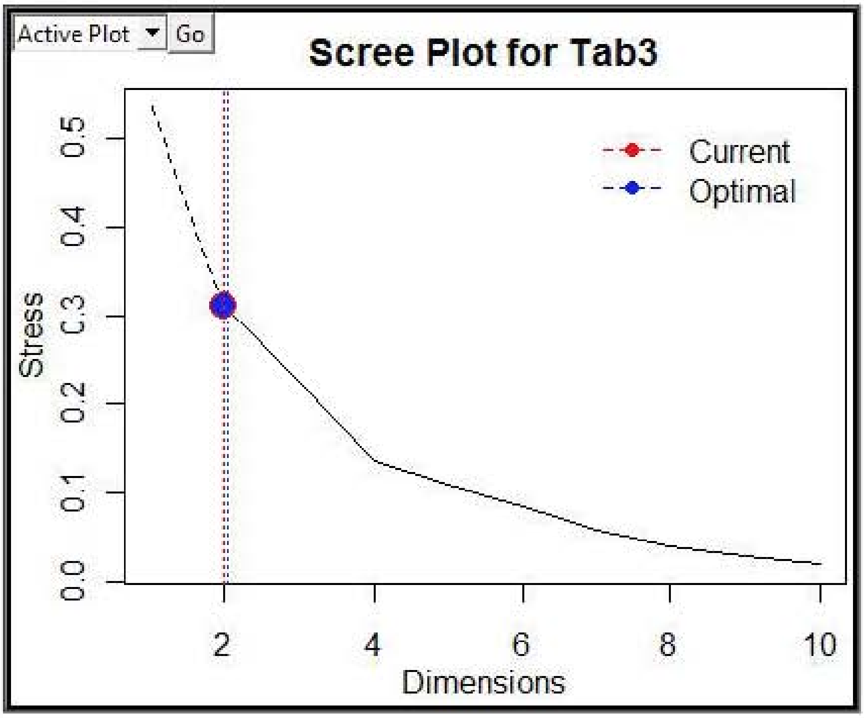


Figure 6.16: Cereal Data: Kruskal's Analysis Scree Plot

The Scree Plot produced for the case of Kruskal's Analysis using the Bray-Curtis distance is shown in Figure 6.16. According to the calculations performed by the MDS-GUI, the optimum dimension is shown to be at dimension 2. Another notable 'kink' in the slope also occurs when $p = 4$, and arguably may be more prominent than that at $p = 2$. Interested researchers may be inclined to extend the analysis to investigate the results when $p = 4$. This however may not be ideal due to the difficulty of visually assessing any configuration with more than three dimensions.

6.3.2 Cereal Data: Configuration Analysis

The configuration produced by Kruskal's Analysis with dissimilarities calculated by the Bray-Curtis measure is shown in Figure 6.17.

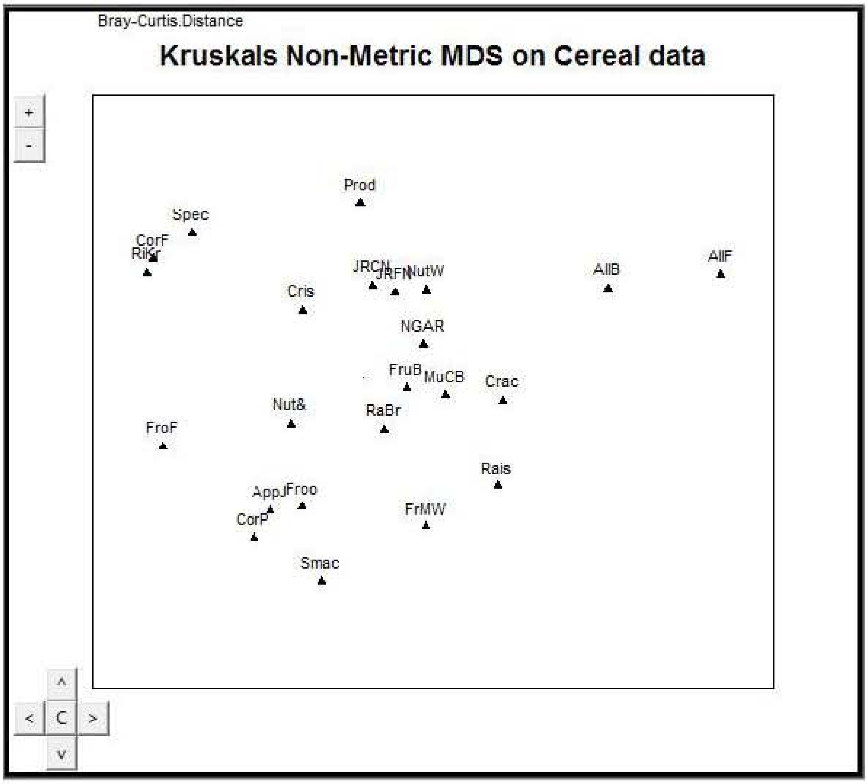


Figure 6.17: Cereal Data: Kruskal's Analysis Configuration

Simple observation of the points does not provide any noteworthy observations, except for the fact that the two objects, *AllB* (All Bran Flakes) and *AllF* (All Bran: Extra Fiber) appear to be the most removed of all the objects. The most obvious distinction of these two cereals, according to the data found in Table A.8, is their high *Dietary Fiber* and *Potassium* content. The configuration can therefore be explained according to the individual variables of the data, to at least some degree. Further analysis of the results will incorporate the displaying of the variable axes through the configuration. There are ten variables that make up the data, and thus ten separate axes are drawn through the origin point of the configuration, as shown in Figure 6.18. Each axis displays markers which provide information regarding the direction and numerical progression of the variable appropriate to the configuration.

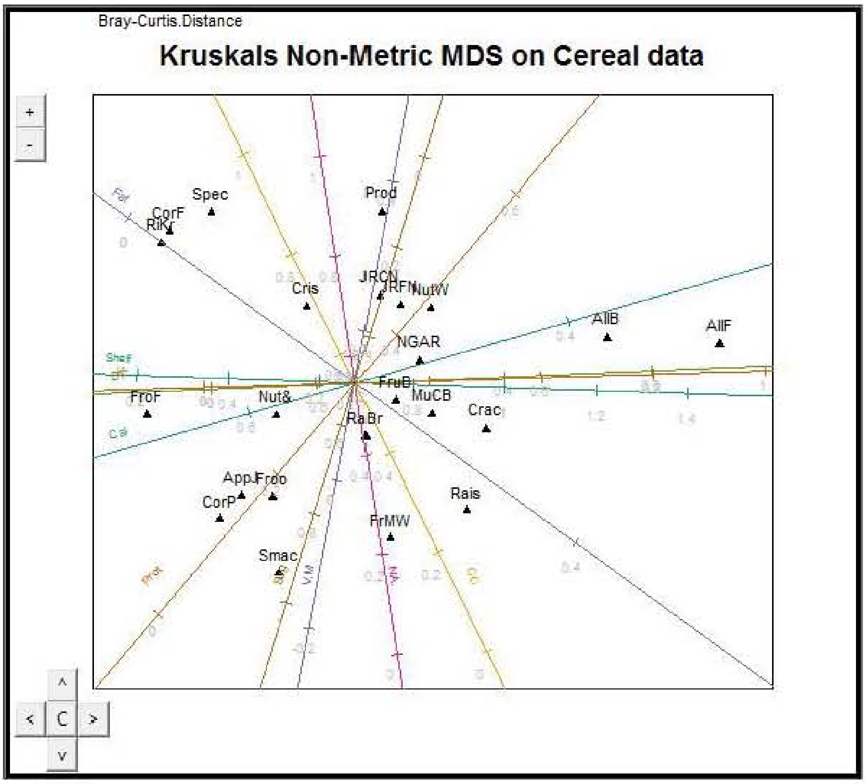


Figure 6.18: Cereal Data: Kruskal's Analysis Configuration (Axes)

Each axis in Figure 6.18 is automatically assigned its own individual colour by the MDS-GUI allowing for easier interpretation. For the most part, the variable axes are well spread out throughout the configuration, which is a typical characteristic of such plots. The researcher is able to make observations about each object with regards to each variable based on where they lie in comparison to the orientation of the axis. For example the *Smac* object (Smacks Cereal) is located at the highest point on the positive direction of the *Sug* (Sugars) variable, which suggests that it has the highest sugar content of all cereals even without having to look at the data. This observation does in fact correspond to the data.

One of the more useful features of analyses with variable axes such as these, is in associating relationships of variables. These observations may not be clear when assessing data, especially when very large, however the relationships are determined visually when added to such configurations. The relationships that will now be commented on are regarding the grouping of variables (grouping defined by sufficiently small angles between axes) that run more or less horizontally through the plot. The four axes in this group correspond to the following variables: *Shelf* (Dark Green-1), *Potassium* (Brown), *Dietary Fiber* (Light Green) and *Calories* (Dark Green-2). The same plot is displayed once again in Figure 6.19, except only the four variable axes in question are provided. From this plot, we are able to see more clearly that the *Shelf*, *Dietary Fiber* and *Potassium* axes all increase from left to right, while the *Calories* axis progresses from right to left. This suggests that there is an inverse relationship between calories and the other three variables. It also suggests that cereals with higher Dietary Fiber are also found to have higher potassium and *vice-versa*. Analysis of the plot also gives a very good idea of the manner in which the shop owner, in the store from which the data was gathered, arranges the cereals on his shelves. A reasonable assumption that one may make about any breakfast cereal is that a cereal with a higher calorie count and lower dietary fiber content would be considered 'unhealthy'. Consequently cereals with high dietary fiber and low calories are considered 'healthy'. Taking the *Shelf* variable into account, it is clear that a higher shelf value is associated with a higher dietary fiber content and a lower calorie count. The shopkeeper therefore arranges the cereals such that the 'unhealthy' cereals are placed on the first shelves and the healthier cereals are placed on the shelves further away. Similar information may be valuable to those interested in the data from a marketing point of view.

The four variables assessed in Figure 6.19 may also be shown to be among the most influential in determining the observed proximities, δ, and therefore in deriving the ordination distances, d. Figure 6.20(b) gives the Shepard Plot with the furthest point (with regards to δ and d) highlighted. Figure 6.20(a) depicts the corresponding pairing with the distance displayed. The two furthest points are *AllF* (All Bran: Extra Fiber) and *FroF* (Frosted Flakes). This pairing is seen to lie on the same range of axes as those of Figure 6.19.

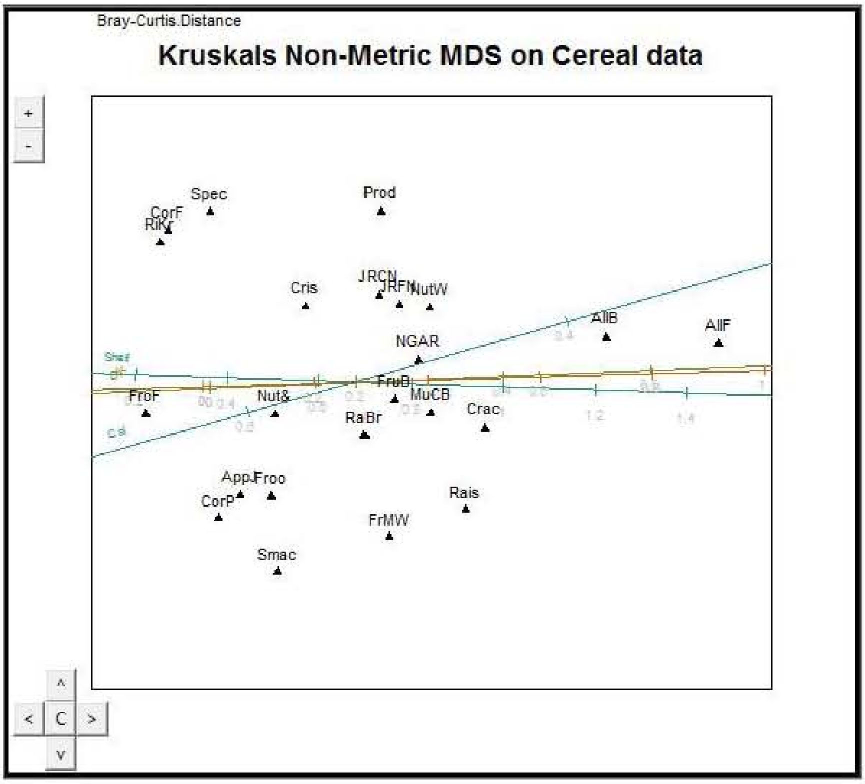
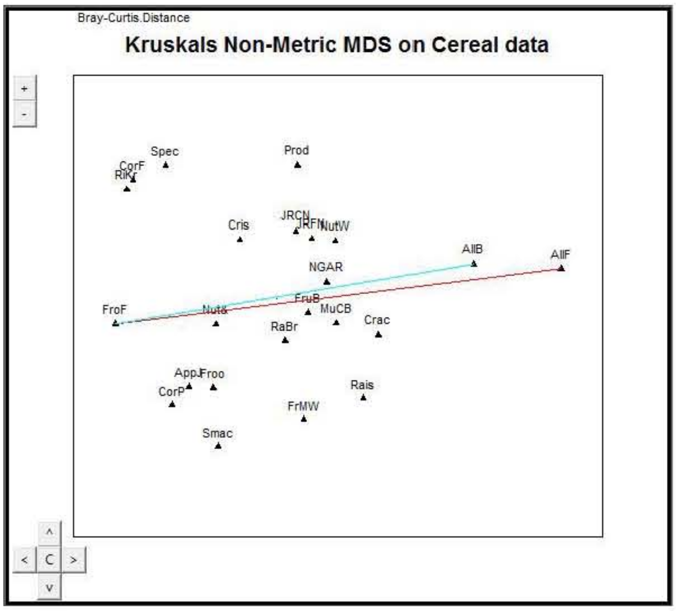
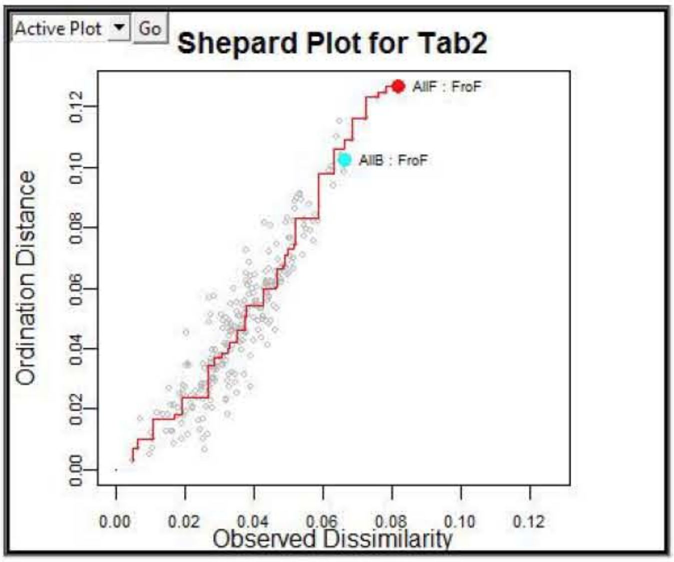


Figure 6.19: Cereal Data: Kruskal's Analysis Configuration (Shelf Axes)

An interesting observation may also be made about the similarities of the appearance of the cereal. Common knowledge of breakfast cereals may suggest that the cereals *AllB* (All Bran Flakes) and *FroF* (Frosted Flakes) have a very similar appearance being that they are both in 'flake' form. According to the results of the Kruskal's Analysis and Figure 6.20 however, the two are very far apart. This goes to show how little appearance effects the perception of similarities of these products in the context of this study.



(a) Cereal Data: Furthest Points (Configuration)



(b) Cereal Data: Furthest Points (Shepard Plot)

Figure 6.20: Cereal Data: Furthest Points

Chapter 7

Conclusions & Recommendations

This final Chapter will serve to summarise the extent and outcomes of the research and software development that was the subject of the dissertation. The Chapter will first describe the significance of the MDS-GUI in terms of its impact on the academic community. It will then discuss the objectives mentioned in Chapter 1 in a concluding manner and the results of the analyses of the real data sets in Chapter 6. Finally some recommendations will be made to those interested in using the MDS-GUI in future research.

7.1 Significance of Research and Development

The first version of the MDS-GUI should be available for public download by the end 2012, where it will have been submitted both to the CRAN database and RForge. The program will be the first complete MDS based GUI for the *R*-Environment, and will also be the package that provides access to the widest range of MDS methods in *R*. The MDS-GUI was developed with both statisticians and non-statisticians as its users in mind.

Statisticians will see the program as a swift means of uploading data and finding a satisfactory MDS result from many options. If this were done from scratch the process would be very time consuming which may be reason for the researcher to abandon MDS as a means of analysis entirely. Additional features of the GUI, such as its ability to perform Procrustes Analysis and display the variable axes like in biplot analysis, also provide the statistical researcher with tools that they may not have considered before using the software. It also provides the more experienced MDS user with the option to expand on the MDS methods that they usually utilise. For example, researchers who before exclusively used SMACOF processes may become aware of the benefits of Kruskal's Analysis easily and without dedicating too much time. Another area where the MDS-GUI will have use is in analysis of industrial data. Research and Development teams in industrial plants may use such software to produce MDS plots on a daily or monthly basis to review plant production. This is something that the MDS-GUI is able to do with only a few clicks of the mouse. The feature which efficiently exports results to PDF format allows for quick production of documents which may be submitted to supervisors or kept for records.

Non-statisticians will hopefully see the MDS-GUI as a piece of software that is more approachable than the coding alternative. People involved in marketing firms, biological studies and doing social science experiments (to name only a few) will all have use of Multidimensional Scaling in at least some of their research and processes. The program will provide these parties with an easy means of producing informative plots, of which they may alter the appearance, in a black box manner. I.e. they are able to simply initiate a process and accept the output without concern of the algorithms and mathematics of the methods at work. After only a brief exploratory session of the MDS-GUI, users will find navigation of the software fairly straightforward and begin producing results from statistical techniques that may never have been available to them before.

7.2 Concluding Remarks About Objectives

- The first objective stated in Chapter 1 was to provide suitable information regarding all aspects of Multidimensional Scaling. Chapters 2 and 3 discussed Multidimensional Scaling from a theoretical point of view and a mathematical explanation of the MDS algorithms, respectively. The theoretical Chapter was set out in such a way that the reader would be able to follow and understand the progression of theory with no prior knowledge of MDS and only a little knowledge of mathematical principals. Chapter 3 was then aimed at a reader with a greater mathematical understanding of multivariate concepts and matrix algebra. The document was designed however, to ensure that the third Chapter only need be applicable to interested readers, and those not familiar with mathematical concepts would be able to skip it and still follow the later chapters.

- The second objective was to provide information of the programming tools used in the development of the MDS-GUI. Similar to Chapter 3, Chapter 4 included information that need only have been read by those interested in how the software was developed. This Chapter provided details of all coding languages and packages used. This simultaneously gave information on vital development tools and gave credit to those who unknowingly contributed to the MDS-GUI.

- Chapter 5 accounts for both the third and fourth objectives. The final product of the MDS-GUI that was developed went above and beyond the initial expectations of the developers in terms of features and interactability. Regularly throughout the course of development, new ideas for features were implemented. The result is a piece of software that provides a diverse range of methods for MDS and MDS result analysis. All aspects of the MDS-GUI are discussed thoroughly in Chapter 5, including navigation and features.

- The final objective, being the demonstration of the MDS-GUI on real data, was accounted for in Chapter 6. The three data sets were chosen to illustrate three separate aspects of MDS results. The Morse-Code data was chosen to show the ability of the MDS-GUI to reproduce well known results; The SynTReN Microarray data to demonstrate the effects of successful category classification and configuration structuring; and the Breakfast Cereal data to show the merits of using the underlying variables of the data as another useful means of configuration analysis.

7.3 Data Study Conclusions

The main observations regarding each of the data studies of Chapter 6 will be briefly summarised here.

- The study of the Morse-Code data using Multidimensional Scaling techniques with the MDS-GUI revealed the following. The MDS method found to be most effective for the data was Non-Metric SMACOF scaling, and produced a Normalised Raw Stress value of 0.039. The non-metric nature of the procedure allowed for a relaxation of the metric assumptions and therefore the produced configuration achieved an optimum result by overstating the majority of the paired object distances. The configuration itself showed clear distinction between groupings of the data based on the length of the Morse-Code sequence of each object. This result strongly suggests that subjects involved in the experiment conducted by Rothkopf (1957) were more inclined to incorrectly identify two sequences as 'the same' when they were of an equal sequence length.

- The SynTReN Microarray data when analysed by the MDS-GUI revealed the following. The results, over all four tested distance metrics, were near identical between Kruskal's Analysis and Non-Metric SMACOF. The method chosen for analysis was Metric SMACOF using a City-Block metric, which produced a Normalised Raw Stress Value of 0.011. The MDS configuration revealed that the structure of the points strongly resembled the 'true' network of the Ecoli provided by the SynTReN software. The two connector genes in the configuration were found to have been placed with the least accuracy, as suggested by the Shepard Diagram, and accounted for the majority of the element of stress. This inadequate placement is suspected to be due to the fact that these genes have the least structural definition according to the 'true' network.

- The study of Breakfast Cereal data produced the following. The analysis progressed using a Kruskal's Analysis with the Bray-Curtis metric, which resulted in a STRESS-2 value of 0.31. The basis of interpretation in this study was with the use of the underlying variable axes of the data. The most prominent observations were with regards to the variables *Shelf Number*, *Potassium*, *Dietary Fiber* and *Calories*. A strong positive correlation was seen to exist among the *Shelf Number*, *Potassium* and *Dietary Fiber* variables; whereas a negative correlation existed between the *Calories* variable and the other three. The conclusion drawn from this was that the cereals were arranged on the shelves according to their perceived level of healthiness.

7.4 Recommendations

A major benefit from the MDS-GUI, and indeed all *R* based packages, is its open source nature. Parties interested in using the **MDSGUI** package may freely download it and distribute it. Regardless of their understanding of MDS and/or coding, it is recommended that any new user read the User Manual which is both available directly from the MDS-GUI and is an Appendix to this document. The intuitive navigation of the developer may be different than what the user might expect, and the User Manual may avoid any potential confusion.

The current version of the GUI is limited to six MDS methods. Future versions of the software are however intended to provide an even wider range. Researchers who would benefit from the inclusion of specific methods or features are urged to contact the developer with request and motivation. Similarly, if a researcher's unique data causes unexpected run time problems in the software they are urged to bring this to the developer's attention so that future versions may eradicate these blind spots. Advanced *R* and *tcltk* coders may also wish to take advantage of the open source nature of the code and make their own allowances manually.

The MDS-GUI was developed using a number of preexisting *R* packages, and is therefore reliant on their functionality. The current version of the MDS-GUI was developed in the 32-bit mode in version 2.13.0 of *R*. It has however been noticed that as new versions of the various packages and versions of *R* are released, the behavior and compatibility between packages and environment differ from when the MDS-GUI was designed. It is important to note that these unexpected changes are out of the MDS-GUI developers hands. The problems are however expected to stabilise in time when later versions of the packages are fully compatible. For the time being, it is recommended that any user of the MDS-GUI use R-2.13.0 in 32-bit mode and use the package versions acknowledged in Chapter 4.

Finally, users wishing to make regular use of the software are recommended to use the program in conjunction with other MDS packages such as the **smacof** *R* package and the X/GGobi software. Features lacking in the MDS-GUI are available in other programs and *vice versa*. Using a range of programs will extend the MDS tools available to the user and allow for a more thorough analysis of their data.

Bibliography

Adler, D. and Murdoch, D. (2011). *rgl: 3D visualization device system (OpenGL)*. R package version 0.92.798.
URL: *http://CRAN.R-project.org/package=rgl*

Allison, D. B., Cui, X., Page, G. P. and Sabripour, M. (2006). Microarray data analysis: from disarray to consolidation and consensus, *Nature Reviews* **7**: 55–65.

Bhattacharyya, A. (1943). On a measure of divergence between two statistical populations defined by their probability distributions, *Bulletin of the Calcutta Mathematical Society* **35**: 99–109.

Bolboaca, S. and Jantschi, L. (2006). Pearson versus Spearman, Kendall's tau correlation analysis on structure-activity relationships of biologic active compounds, *Leonardo J. Sci* **9**: 179–200.

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

Bray, J. R. and Curtis, J. T. (1957). An ordination of the upland forest communities of southern Wisconsin, *Ecological Monographies* **27**: 325–349.

Brewer, C. and Harrower, M. (2002). Colorbrewer2.0.
URL: *http://colorbrewer2.org*

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

Canty, A. and Ripley, B. (2010). *boot: Bootstrap R(S-Plus) functions.* R package version 1.2-43.
URL: *http://CRAN.R-project.org/package=boot*

Carrol, J. D. and Wish, M. (2002). Multimensional scaling: Models, methods, and relations to delphi, *Murray Turnoff and Harold. A. Linstone.*

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.

Carroll, J. D., Green, P. E. and Carmone, F. J. (1976). CANDELING: A new method for multidimensional scaling with constrained solutions, *Paper presented at the Internation Congress of Psychology. Paris.*

Causton, H. C., Quackenbush, J. and Brazma, A. (2003). *Microarray Gene Expression Data Analysis: A Beginners Guide*, Blackwell Publishing Company, Oxford.

Cha, S. H. (2007). Comprehensive survey on distance/similarity measures between probability density functions, *International Journal of Mathematical Models and Methods in Applied Sciences* **1(4)**: 300–307.

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

Clark, C. (2005). *An Introduction to Ordination.* San Francisco State University, San Francisco.

Corradino, C. (1990). Proximity structure in a captive colony of japanese monkeys (macaca fuscata fuscata): An application of multidimensional scaling, *Primates* **31**: 351–362.

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

Davison, A. C. and Hinkley, D. V. (1997). Bootstrap methods and their applications, *Cambridge University Press, Cambridge .*

de Leeuw, J. and Heiser, W. J. (1982). Theory of multidimensional scaling. in P. R. Krishnaiah and L. N. Kanl (eds.), *Handbook of statistics* **2**: 285–316.

de Leeuw, J. and Mair, P. (2009a). Gifi methods for optimal scaling in R: The package homals, *Journal of Statistical Software* **31**(4): 1–20.
URL: *http://www.jstatsoft.org/v31/i04/*

de Leeuw, J. and Mair, P. (2009b). Multidimensional scaling using majorization: SMACOF in R, *Journal of Statistical Software* **31**(3): 1–30.
URL: *http://www.jstatsoft.org/v31/i03/*

Eckart, C. and Young, G. (1936). The approximation of one matrix by another of lower rank, *Psychometrika* **1**: 211–218.

Evans, M. W. (2010). *Equations of motion from the Minkowski metric.*
URL: *www.aias.com*

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

Fawcett, C. D. (1901). A second study of the variation and correlation of the human skull, with special reference to the naqada crania, *Biometrika* **1**: 408–467.

Gower, J. C. (1966). Some distance properties of latent root and vector methods used in multivariate analysis, *Biometrika* **53**: 325–338.

Gower, J. C. and Hand, D. J. (1996). *Biplots*, Chapman and Hall, London.

Gower, J., Lubbe, S. and le Roux, N. (2011). *Understanding Biplots*, John Wiley, Chichester.

Greenacre, M. (2007). *Correspondence Analysis in Practice, Second Edition*, Chapman and Hall/CRC, London.

Groenen, P. J. F. (2003). Interactive multidimensional scaling iMDS v0.1. A standalone Windows application (XP, Vista) that allows dynamic form multidimensional scaling.
URL: *http://people.few.eur.nl/groenen/*

Groenen, P. J. F. and Heiser, W. J. (1996). The tunneling method for global optimization in multidimensional scaling, *Psychometrika* **61**: 529–550.

Groenen, P. J. F. and van de Velden, M. (2004). Multidimensional scaling, *Technical Report EI-2004-15*, Erasmus University, Rotterdam.

Grosjean, P. (2011). *SciViews-R: A GUI API for R.*
URL: *http://www.sciviews.org/SciViews-R*

Guttman, L. (1968). A general nonmetric technique for finding the smallest coordinate space for a configuration of points, *Psychometrika* **33**: 469–506.

Heiser, W. J. (1988). The city-clock model for three-way multidimensional scaling, *Technical report*, Department of Data, University of Leiden, Leiden.

Heiser, W. J. and Meulman, J. J. (1983). Analyzing rectangular tables by joint and constrained MDS, *Journal of Econometrics* **22**: 193–167.

Hirschfeld, H. O. (1935). A connection between correlation and contingency, *Mathematical Proceedings of the Cambridge Philosophical Society* **31**: 520–524.

Holland, S. M. (2008). Non-metric multidimensional scaling, *Technical Report GA 30602-2501*, Department of Geology, University of Georgia, Athens.

Hossain, M. S., Naryan, M. and Ramakrishnan, N. (2010). Efficiency discovering hammock paths from induced similarity networks, *Technical report*, Department of Computer Science, Virginia Tech, Blacksburg.

Hotelling, H. (1936). Analysis of a complex of statistical variables into principal components, *Journal of Educational Psychology* **24**: 417–441.

Husson, F., Le, S. and Cadoret, M. (2011). *SensoMineR: Sensory data analysis with R.* R package version 1.14.
URL: *http://CRAN.R-project.org/package=SensoMineR*

Johnson, R. and Wicheren, D. (2007). *Applied Multivariate Statistical Analysis, International Edition 6*, Prentice-Hall, New York.

Jurman, G., Riccadonna, S., Visintainer, R. and Furlanello, C. (2009). Canberra distance on ranked lists, *Proceedings of Advances in Ranking NIPS 09 Workshop* pp. 22–29.

Kamiechetty, H. M., Natarajan, P. and Rakshit, S. (2002). An empirical framework to evaluate performance of dissimilariy metrics in content-based retrieval systems, *Technical report*, Center for Artificial intelligence and Robotics, Bangalore.

Kruskal, J. B. (1964). Multidimensional scaling by optimising goodness of fit to a nonmetric hypothesis, *Psychometrika* **29**: 1–27,115–129.

Kruskal, J. B. and Wish, M. (1978). *Multidimensional Scaling*, Sage Publications, London.

la Grange, A. M., le Roux, N. J. and Gardner-Lubbe, S. (2009). BiplotGUI: Interactive biplots in R, *Journal of Statistical Software* **30**(12): 1–37.
URL: *http://www.jstatsoft.org/v30/i12*

Lance, G. N. and Williams, W. T. (1966). Computer programs for hierarchical polythetic classification, *Computer Journal* **9**.

Lapointe, F. J. and Lengendre, P. (1994). A classification of pure malt scotch whiskies, **43**(1): 237–257.

le Roux, N. J. (2012). SMACOF R code for metric and non-metric algorithms, *Personal Communication* .

Leisch, F. (2002). Sweave: Dynamic generation of statistical reports using literate data analysis, *in* W. Härdle and B. Rönz (eds), *Compstat 2002 — Proceedings in Computational Statistics*, Physica Verlag, Heidelberg, pp. 575–580.
URL: *http://www.stat.uni-muenchen.de/ leisch/Sweave*

Lewis, R. M. and Trosset, M. W. (2009). Extensions of classical multidimensional scaling: Variable alternation and nonconvex duality, *Technical report*, Williamsburg, Virginia.
URL: *http://www.scientificcommons.org/42628323*

Ligges, U. and Mächler, M. (2003). Scatterplot3d - an R package for visualizing multivariate data, *Journal of Statistical Software* **8**(11): 1–20.
URL: *http://www.jstatsoft.org*

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*

Mahalanobis, P. (1936). On the generalised distance in statistics, *Proceedings of the National Institute of Science of India* **2**: 49–55.

Mair, P. and de Leeuw, J. (2008). Multidimensional scaling using majorization: SMACOF in R, *Technical report*, Department of Statistics, UCLA, Los Angeles.

Markos, A. (2010). *caGUI: A Tcl/Tk GUI for the functions in the ca package*. R package version 0.1-4.
URL: *http://CRAN.R-project.org/package=caGUI*

Martin, S. (1993). Effective visual communication for graphical user interfaces.
URL: *http://web.cs.wpi.edu/ matt/courses/cs563/talks/smartin/*

Neuwirth, E. (2011). *RColorBrewer: ColorBrewer palettes*. R package version 1.0-5.
URL: *http://CRAN.R-project.org/package=RColorBrewer*

Oksanen, J., Blanchet, F. G., Kindt, R., Legendre, P., O'Hara, R. B., Simpson, G. L., Solymos, P., Stevens, M. H. H. and Wagner, H. (2011). *vegan: Community Ecology Package*. R package version 1.17-8.
URL: *http://CRAN.R-project.org/package=vegan*

Ousterhout, J. K. and Jones, K. (2010). *Tcl and the Tk toolkit*, Pearson Eductation, London.

Pearson, K. (1901). On lines and planes of closest fit to a system of points in space, *Philosophical Magazine* **2**: 557–572.

R Development Core Team (2012). *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria.
URL: *http://www.R-project.org/*

Roberts, D. W. (2010). *labdsv: Ordination and Multivariate Analysis for Ecology.* R package version 1.4-1.
URL: *http://CRAN.R-project.org/package=labdsv*

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/*

Sambamoorth, N. (2012). Hierarchical cluster analysis: Some basics and algorithms. CRMportals Inc.

Sammon, J. W. (1969). A nonlinear mapping for data structure analysis, *IEEE trans, Comput.* **18**: 401–409.

SAS Institute Inc. (2011). *Base SAS 9.3 procedures guide.* NC: SAS Institute Inc.
URL: *www.sas.com*

Schulz, J. (2007). Bray-curtis dissimilarity. code10.info: A Developing variety of technical and computational topics from different scientific domains.
URL: *http://www.code10.info*

SGI (2012). Opengl software development kit.
URL: *http://www.opengl.org*

Shepard, R. N. (1962). The analysis of proximities: multidimensional scaling with an unknown distance function, *Psychometrika* **27**: 125–140.

Silva, J. and Narayanan, S. (2006). Average divergence distance as a statistical discrimination measure for hidden markov models, *IEEE Transactions on speech and audio processing* **14**: 890–906.

Smyth, G. K. (2005). *Limma: linear models for microarray data. In: Bioinformatics and Computational Biology Solutions using R and Bioconductor*. Bioconductor.
URL: *http://www.bioconductor.org/packages/2.10/bioc/html/limma.html*

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* **43**: 423–444.

Ter Braak, C. J. F. (1992). Multidimensional scaling and regression, *Psychological Review* **34**: 273–286.

Tierney, L. (2011). *tkrplot: TK Rplot*. R package version 0.0-20.
URL: *http://CRAN.R-project.org/package=tkrplot*

Torgerson, W. S. (1952). Multidimensional scaling: I. theory and method, *Psychometrika* **17**: 401–419.

Van den Bulcke, T., Van Leemput, K., Naudts, B., van Remortel, P., Ma, H., Verschoren, A., De Moor, B. and Marchal, K. (2006). Syntren: a generator of synthetic gene expression data for design and analysis of structure learning algorithms, *BMC Bioinformatics* **7**.

Venables, W. N. and Ripley, B. D. (2002). *Modern Applied Statistics with S, Fourth edition*, Springer, New York.
URL: *http://www.stats.ox.ac.uk/pub/MASS4*

Wickelmaier, F. (2003). *An introduction to MDS*. Sound Quality Research Unit, Aalborg University, Denmark.

Young, F. W. (1985). Multidimensional scaling, *Johnson and Kotz Encyclopedia of Statistical Sciences* **5**. University of North Carolina, North Carolina.

Young, G. and Householder, A. S. (1938). Discussion of a set of points in terms of their mutual distances, *Psychometrika* **3**: 19–22.

Appendix A

Data Sets

Four data sets were used during the course of the dissertation. Each of these will be described in this Appendix Chapter.

A.1 Skulls Data

The skulls data set, used for demonstration in Chapters 2 and 5, forms a subset of the data gathered by Pearson (1901) and published by Fawcett (1901). The original data comprised 48 measurements on 100 ancient Egyptian human skulls. The subset of this data, used in this dissertation and in Cox and Cox (2001), takes only twelve of the measurements and selects only 40 of the skulls. Of this 40, 22 skulls are identified as male and 18 as female, with the distinction identifiable by the name. Table A.1 lists the twelve variables (measurements) and their abbreviated name. Table A.2 then gives the skulls data set in the format uploaded into, and used by, the MDS-GUI.

Table A.1: Skull Variables

Variable		**Variable**	
Greatest Length	L	Breadth	B
height	H	Auricular height	OH
Circumference	U	Sagittal Circumference	S
Cross-Circumference	Q	Upper Face Height	G'H
Nasal Breadth	NB	Nasal Height	NH
Cephalic Index	B/L	Height/Length	H/L

Table A.2: Skulls Data

	L	**B**	**H**	**OH**	**U**	**S**	**Q**	**G'H**	**NB**	**NH**	**B/L**	**H/L**
5F	-2.27	-0.781	-1.74	-1.33	-1.89	-0.902	-1.48	-1.9	-1.41	-1.06	1.412	0.496
7M	0.0364	0.3764	-0.966	-0.685	0.3166	-0.652	-0.922	0.1855	-0.453	0.4895	0.2161	-1.12
10F	-0.707	1.557	-0.327	0.1476	-0.418	-0.401	1.124	0.511	0.6635	-0.8	1.718	0.3807
13M	0.928	1.439	-0.675	-0.096	1.261	0.1838	1.557	1.357	2.577	0.7475	0.2161	-1.66
26M	1.671	1.676	1.163	1.621	2.171	1.52	2.053	0.4459	0.9187	0.2315	-0.213	-0.465
32M	-0.558	0.8488	-1.93	-1.07	-0.278	-1.4	-0.364	-0.856	-0.293	1.263	1.074	-1.58
43F	-1.45	0.2583	-1.55	-1.01	-1.01	-1.15	-0.488	-0.596	-1.09	-0.026	1.442	-0.196
45F	-0.409	-0.45	-1.45	-2.63	-1.05	-1.57	-1.11	-0.596	-0.772	-2.09	0.0015	-1.2
46F	-1.38	1.085	-0.675	-0.173	-0.208	-0.401	0.8756	-1.77	-0.453	-0.284	1.994	0.7268
52M	-0.558	-0.096	-0.579	-0.173	-0.313	-0.401	-0.116	0.4459	-0.134	0.7475	0.3694	-0.081
58F	0.2593	-1.04	0.1955	0.1476	0.3516	0.1838	-0.24	1.227	1.142	0.7475	-0.979	-0.081
59F	-0.112	-2.06	-0.772	0.7241	-0.733	-0.067	-0.612	-1.51	-1.41	0.2315	-1.41	-0.734
63F	0.5565	0.7307	0.4858	1.236	0.8414	1.604	1.247	-1.25	-0.772	0.2315	0.0322	-0.081
64F	-1.75	-1.04	-0.579	-1.84	-1.75	-2.53	-2.35	-1.25	-0.453	-1.32	0.5839	1.265
66M	-0.409	1.203	-0.966	0.0836	-0.068	-1.49	0.8756	1.227	0.6635	2.295	1.197	-0.658
70F	-1.87	-0.923	-0.385	-0.493	-1.96	-1.15	-1.11	-1.12	-2.37	-0.026	0.9211	1.611
83F	1.077	1.557	-0.385	0.2117	1.191	1.103	0.3797	0.4459	1.381	-0.8	0.1854	-1.5
85M	0.8537	0.8488	0.1955	-0.557	0.9464	0.3091	-0.364	-1.64	-0.772	-0.8	-0.121	-0.658
86M	-0.186	0.0221	0.9696	0.3398	-0.033	0.3091	0.1317	0.7062	-0.293	-0.8	0.1548	1.265
93bM	-1.15	0.6126	-1.35	-1.07	-0.558	-1.24	0.2557	-0.465	0.6635	-1.06	1.442	-0.311
96M	1.002	1.321	1.744	1.749	1.261	1.771	2.115	-0.596	-0.453	0.7475	0.0935	0.8037
97F	-1.6	1.085	0.0019	0.1476	-1.36	-0.735	-0.116	0.5761	0.6635	0.8764	2.209	1.765
99M	0.928	0.3764	0.9696	1.365	0.6665	0.6015	0.7516	-0.205	-1.25	-1.06	-0.52	0.0731
102M	0.185	0.1402	0.1955	-0.813	-0.488	-0.15	-0.488	0.4459	1.142	-1.32	-0.06	-0.004
112M	0.4822	-2.22	0.8342	-0.045	-0.488	-0.234	-0.86	0.6411	1.302	-0.284	-1.96	0.3807
120M	2.191	0.4945	0.9696	0.2117	2.451	0.8521	0.5036	1.748	1.302	2.295	-1.38	-1.16
121F	0.0364	-0.45	-1.16	-1.33	-0.418	-0.902	-0.86	-0.335	-0.453	-0.284	-0.366	-1.31
125M	-0.261	-0.686	1.647	1.236	-0.313	0.1838	0.9995	-0.856	-0.612	-1.45	-0.305	2.073
136M	1.225	-1.04	-0.385	0.7241	0.8414	1.103	-0.116	1.097	0.5838	-0.026	-1.72	-1.66
137F	0.185	-0.923	-0.385	-0.557	-0.208	0.3091	-1.23	1.487	0.1053	0.2315	-0.826	-0.619
138M	0.3336	-0.332	-0.192	0.2117	0.0367	0.1838	-0.116	-0.856	-0.772	0.7475	-0.52	-0.581
139M	0.0364	-0.923	-0.385	0.0836	-0.698	-0.067	-0.364	-0.335	0.504	1.005	-0.703	-0.465
140F	0.7794	0.0221	0.9696	1.749	0.7714	0.9356	1.619	0.7062	0.823	-1.83	-0.642	0.2269
143M	0.185	0.0221	0.7761	-0.429	0.1067	0.1002	-0.116	0.3157	-0.214	-0.542	-0.152	0.6499
144F	1.077	-0.521	0.9696	1.493	0.9114	1.144	0.2557	0.8364	-0.772	1.005	-1.22	-0.081
145F	0.185	-0.096	1.744	0.7241	0.1417	1.855	0.5036	-0.075	0.504	0.6443	-0.244	1.688
146F	0.0364	-0.509	0.389	-1.33	-0.908	0.1002	-1.48	0.7062	-0.772	-0.155	-0.428	0.3807
148M	0.185	-1.75	0.2922	0.3398	0.0717	-0.15	-0.488	-0.856	-0.772	-0.026	-1.41	0.1115
151M	-1	-0.805	0.389	0.0836	-0.628	-0.15	-0.86	-0.075	0.0255	0.2315	0.2468	1.496
152M	1.225	0.9669	1.937	0.9803	1.436	1.395	0.9995	2.008	1.78	1.263	-0.336	0.8037

A.2 Morse-Code Data

Morse-Code is a universal, non spoken, means of transmitting messages. The code uses a series of long and short ‘beeps’ where every letter and number has its own sequence. These long and short signals are described as ‘dashes’ and ‘dots’ respectively. Table A.3 gives the code sequences for each letter of the alphabet and number from zero to nine.

Table A.3: Morse Code Symbols

Symbol	**Code**	**Symbol**	**Code**	**Symbol**	**Code**
A	.-	**M**	- -	**Y**	-.- -
B	-...	**N**	-.	**Z**	- -..
C	-.-.	**O**	- - -	**1**	.- - - -
D	-..	**P**	.- -.	**2**	..- - -
E	.	**Q**	- -.-	**3**	...- -
F	..-.	**R**	.-.	**4**	....-
G	- -.	**S**	...	**5**	.....
H	....	**T**	-	**6**	-....
I	..	**U**	..-	**7**	- -...
J	.- - -	**V**	...-	**8**	- - -..
K	-.-	**W**	.- -	**9**	- - - -.
L	.-..	**X**	-..-	**0**	- - - - -

The data collected by Rothkopf (1957) has become well known and somewhat associated with Multidimensional Scaling. The study set out to obtain confusion data by asking 598 subjects to identify whether two signals, played consecutively, were the same or not. Each subject were presented each of the 36^2 pairs of signals. The data shown in Table A.4 gives the asymmetric results, where each element gives the percentage of subjects that perceived the pairing to be the same. The data is thus in the form of the $n \times n$ similarity matrix.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	N1	N2	N3	N4	N5	N6	N7	N8	N9	N0
A	0.92	0.04	0.06	0.13	0.03	0.14	0.1	0.13	0.46	0.05	0.22	0.03	0.25	0.34	0.06	0.06	0.09	0.35	0.23	0.06	0.37	0.13	0.17	0.12	0.07	0.03	0.02	0.07	0.05	0.05	0.08	0.06	0.05	0.06	0.02	0.03
B	0.05	0.84	0.37	0.31	0.05	0.28	0.17	0.21	0.05	0.19	0.34	0.4	0.06	0.1	0.12	0.22	0.25	0.16	0.18	0.02	0.18	0.34	0.08	0.84	0.3	0.42	0.12	0.17	0.14	0.4	0.32	0.74	0.43	0.17	0.04	0.04
C	0.04	0.38	0.87	0.17	0.04	0.29	0.13	0.07	0.11	0.19	0.24	0.35	0.14	0.03	0.09	0.51	0.34	0.24	0.14	0.06	0.06	0.11	0.14	0.32	0.82	0.38	0.13	0.15	0.31	0.14	0.1	0.3	0.28	0.24	0.18	0.12
D	0.08	0.62	0.17	0.88	0.07	0.23	0.4	0.36	0.09	0.13	0.81	0.56	0.08	0.07	0.09	0.27	0.09	0.45	0.29	0.06	0.17	0.2	0.27	0.4	0.15	0.33	0.03	0.09	0.06	0.11	0.09	0.19	0.08	0.1	0.05	0.06
E	0.06	0.13	0.14	0.06	0.97	0.02	0.04	0.04	0.17	0.01	0.05	0.06	0.04	0.04	0.05	0.01	0.05	0.1	0.07	0.67	0.03	0.03	0.02	0.05	0.06	0.05	0.04	0.03	0.05	0.03	0.05	0.02	0.04	0.02	0.03	0.03
F	0.04	0.51	0.33	0.19	0.02	0.9	0.1	0.29	0.05	0.33	0.16	0.5	0.07	0.06	0.1	0.42	0.12	0.35	0.14	0.02	0.21	0.27	0.25	0.19	0.27	0.13	0.08	0.16	0.47	0.25	0.26	0.24	0.21	0.05	0.05	0.05
G	0.09	0.18	0.27	0.38	0.01	0.14	0.9	0.06	0.05	0.22	0.33	0.16	0.14	0.13	0.62	0.52	0.23	0.21	0.05	0.03	0.15	0.14	0.32	0.21	0.23	0.39	0.15	0.14	0.05	0.1	0.04	0.1	0.17	0.23	0.2	0.11
H	0.03	0.45	0.23	0.25	0.09	0.32	0.08	0.87	0.1	0.1	0.09	0.29	0.05	0.08	0.08	0.14	0.08	0.17	0.37	0.04	0.36	0.59	0.09	0.33	0.14	0.11	0.03	0.09	0.15	0.43	0.7	0.35	0.17	0.04	0.03	0.03
I	0.64	0.07	0.07	0.13	0.1	0.08	0.06	0.12	0.93	0.03	0.05	0.16	0.13	0.3	0.07	0.03	0.05	0.19	0.35	0.16	0.1	0.05	0.08	0.02	0.05	0.07	0.02	0.05	0.08	0.09	0.06	0.08	0.05	0.02	0.04	0.05
J	0.07	0.09	0.38	0.09	0.02	0.24	0.18	0.05	0.04	0.85	0.22	0.31	0.08	0.03	0.21	0.63	0.47	0.11	0.02	0.07	0.09	0.09	0.09	0.22	0.32	0.28	0.67	0.66	0.33	0.15	0.07	0.11	0.28	0.29	0.26	0.23
K	0.05	0.24	0.38	0.73	0.01	0.17	0.25	0.11	0.05	0.27	0.91	0.33	0.1	0.12	0.31	0.14	0.31	0.22	0.02	0.02	0.23	0.17	0.33	0.63	0.16	0.18	0.05	0.09	0.17	0.08	0.08	0.18	0.14	0.13	0.05	0.06
L	0.02	0.69	0.43	0.45	0.1	0.24	0.12	0.26	0.09	0.3	0.27	0.86	0.06	0.02	0.09	0.37	0.36	0.28	0.12	0.05	0.16	0.19	0.2	0.31	0.25	0.59	0.12	0.13	0.17	0.15	0.26	0.29	0.36	0.16	0.07	0.03
M	0.24	0.12	0.05	0.14	0.07	0.17	0.29	0.08	0.08	0.11	0.23	0.08	0.96	0.62	0.11	0.1	0.15	0.2	0.07	0.09	0.13	0.04	0.21	0.09	0.18	0.08	0.05	0.07	0.06	0.06	0.05	0.07	0.11	0.07	0.1	0.04
N	0.31	0.04	0.13	0.3	0.08	0.12	0.1	0.16	0.13	0.03	0.16	0.08	0.59	0.93	0.05	0.09	0.05	0.28	0.12	0.1	0.16	0.04	0.12	0.04	0.16	0.11	0.05	0.02	0.03	0.04	0.04	0.06	0.02	0.02	0.1	0.02
O	0.07	0.07	0.2	0.06	0.05	0.09	0.76	0.07	0.02	0.39	0.26	0.1	0.04	0.08	0.86	0.37	0.35	0.1	0.03	0.04	0.11	0.14	0.25	0.35	0.27	0.27	0.19	0.17	0.07	0.07	0.06	0.18	0.14	0.11	0.2	0.12
P	0.05	0.22	0.33	0.12	0.05	0.36	0.22	0.12	0.03	0.78	0.14	0.46	0.05	0.06	0.21	0.83	0.43	0.23	0.09	0.04	0.12	0.19	0.19	0.19	0.41	0.3	0.34	0.44	0.24	0.11	0.15	0.17	0.24	0.23	0.25	0.13
Q	0.08	0.2	0.38	0.11	0.04	0.15	0.1	0.05	0.02	0.27	0.23	0.26	0.07	0.06	0.22	0.51	0.91	0.11	0.02	0.03	0.06	0.14	0.12	0.37	0.5	0.63	0.34	0.32	0.17	0.12	0.09	0.27	0.4	0.58	0.37	0.24
R	0.13	0.14	0.16	0.23	0.05	0.34	0.26	0.15	0.07	0.12	0.21	0.33	0.14	0.12	0.12	0.29	0.08	0.87	0.16	0.02	0.23	0.23	0.62	0.14	0.12	0.13	0.07	0.1	0.13	0.04	0.07	0.12	0.07	0.09	0.01	0.02
S	0.17	0.24	0.05	0.3	0.11	0.26	0.05	0.59	0.16	0.03	0.13	0.1	0.05	0.17	0.06	0.06	0.03	0.18	0.96	0.09	0.56	0.24	0.12	0.1	0.06	0.07	0.08	0.02	0.02	0.15	0.28	0.09	0.05	0.05	0.05	0.02
T	0.13	0.1	0.01	0.05	0.46	0.03	0.06	0.06	0.14	0.06	0.14	0.07	0.06	0.05	0.06	0.11	0.04	0.04	0.07	0.96	0.08	0.05	0.04	0.02	0.02	0.06	0.05	0.05	0.03	0.03	0.03	0.08	0.07	0.06	0.14	0.06
U	0.14	0.29	0.12	0.32	0.04	0.32	0.11	0.34	0.21	0.07	0.44	0.32	0.11	0.13	0.06	0.2	0.12	0.4	0.51	0.06	0.93	0.57	0.34	0.17	0.09	0.11	0.06	0.06	0.16	0.34	0.1	0.09	0.09	0.07	0.04	0.03
V	0.05	0.17	0.24	0.16	0.09	0.29	0.06	0.39	0.05	0.11	0.26	0.43	0.04	0.01	0.09	0.17	0.1	0.17	0.11	0.06	0.32	0.92	0.17	0.57	0.35	0.1	0.1	0.14	0.28	0.79	0.44	0.36	0.25	0.1	0.01	0.05
W	0.09	0.21	0.3	0.22	0.09	0.36	0.25	0.15	0.04	0.25	0.29	0.18	0.15	0.06	0.26	0.2	0.25	0.61	0.12	0.04	0.19	0.2	0.86	0.22	0.25	0.22	0.1	0.22	0.19	0.16	0.05	0.09	0.11	0.06	0.03	0.07
X	0.07	0.64	0.45	0.19	0.03	0.28	0.11	0.06	0.01	0.35	0.5	0.42	0.1	0.08	0.24	0.32	0.61	0.1	0.12	0.03	0.12	0.17	0.21	0.91	0.48	0.26	0.12	0.2	0.24	0.27	0.16	0.57	0.29	0.16	0.17	0.06
Y	0.09	0.23	0.62	0.15	0.04	0.26	0.22	0.09	0.01	0.3	0.12	0.14	0.05	0.06	0.14	0.3	0.52	0.05	0.07	0.04	0.06	0.13	0.21	0.44	0.86	0.23	0.26	0.44	0.4	0.15	0.11	0.26	0.22	0.33	0.23	0.16
Z	0.03	0.46	0.45	0.18	0.02	0.22	0.17	0.1	0.07	0.23	0.21	0.51	0.11	0.02	0.15	0.59	0.72	0.14	0.04	0.03	0.09	0.11	0.12	0.36	0.42	0.87	0.16	0.21	0.27	0.09	0.1	0.25	0.66	0.47	0.15	0.15
N1	0.02	0.05	0.1	0.03	0.03	0.05	0.13	0.04	0.02	0.29	0.05	0.14	0.09	0.07	0.14	0.3	0.28	0.09	0.04	0.02	0.03	0.12	0.14	0.17	0.19	0.22	0.84	0.63	0.13	0.08	0.1	0.08	0.19	0.32	0.57	0.55
N2	0.07	0.14	0.22	0.05	0.04	0.2	0.13	0.03	0.25	0.26	0.09	0.14	0.02	0.03	0.17	0.37	0.28	0.06	0.05	0.03	0.06	0.1	0.11	0.17	0.3	0.13	0.62	0.89	0.54	0.2	0.05	0.14	0.2	0.21	0.16	0.11
N3	0.03	0.08	0.21	0.05	0.04	0.32	0.06	0.12	0.02	0.23	0.06	0.13	0.05	0.02	0.05	0.37	0.19	0.09	0.07	0.06	0.04	0.16	0.06	0.22	0.25	0.12	0.18	0.64	0.86	0.31	0.23	0.41	0.16	0.17	0.08	0.1
N4	0.06	0.19	0.19	0.12	0.08	0.25	0.14	0.16	0.07	0.21	0.13	0.19	0.03	0.03	0.02	0.17	0.29	0.11	0.09	0.03	0.17	0.55	0.08	0.37	0.24	0.03	0.05	0.26	0.44	0.89	0.42	0.44	0.32	0.1	0.03	0.03
N5	0.08	0.45	0.15	0.14	0.02	0.45	0.04	0.67	0.07	0.14	0.04	0.41	0.02	0	0.04	0.13	0.07	0.09	0.27	0.02	0.14	0.45	0.07	0.45	0.1	0.1	0.14	0.1	0.3	0.69	0.9	0.42	0.24	0.1	0.06	0.05
N6	0.07	0.8	0.3	0.17	0.04	0.23	0.04	0.14	0.02	0.11	0.11	0.27	0.06	0.02	0.07	0.16	0.3	0.11	0.14	0.03	0.12	0.3	0.09	0.58	0.38	0.39	0.15	0.14	0.26	0.24	0.17	0.88	0.69	0.14	0.05	0.14
N7	0.06	0.33	0.22	0.14	0.05	0.25	0.06	0.04	0.06	0.24	0.13	0.32	0.07	0.06	0.07	0.36	0.39	0.12	0.06	0.02	0.03	0.13	0.09	0.3	0.3	0.5	0.22	0.29	0.18	0.15	0.12	0.61	0.85	0.7	0.2	0.13
N8	0.03	0.23	0.4	0.06	0.03	0.15	0.15	0.06	0.02	0.33	0.1	0.14	0.03	0.06	0.14	0.12	0.45	0.02	0.06	0.04	0.06	0.07	0.05	0.24	0.35	0.5	0.42	0.29	0.16	0.16	0.09	0.3	0.6	0.89	0.61	0.26
N9	0.03	0.14	0.23	0.03	0.01	0.06	0.14	0.05	0.02	0.3	0.06	0.07	0.16	0.11	0.1	0.31	0.32	0.05	0.06	0.07	0.06	0.03	0.08	0.11	0.21	0.24	0.57	0.39	0.09	0.12	0.04	0.11	0.42	0.56	0.91	0.78
N0	0.09	0.03	0.11	0.02	0.05	0.07	0.14	0.04	0.05	0.3	0.08	0.03	0.02	0.03	0.25	0.21	0.29	0.02	0.03	0.04	0.05	0.03	0.02	0.12	0.15	0.2	0.5	0.26	0.09	0.11	0.05	0.22	0.17	0.52	0.81	0.94

Table A.4: Asymmetric Morse-Code Data

Many Multidimensional Scaling procedures assume that the similarity/ dissimilarity matrix is symmetric. The adapted data is shown in Table A.5. Each element of the symmetric similarity matrix is $\frac{x_{ij}+x_{ji}}{2}$ of the asymmetric data. The diagonal scores have been made zero as many MDS procedures also assume the dissimilarity of an object with itself to be zero. The format of the data in Table A5 is in the form which it is uploaded and used by the MDS-GUI.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	N1	N2	N3	N4	N5	N6	N7	N8	N9	N0
A	0	0.05	0.05	0.11	0.05	0.09	0.1	0.08	0.55	0.06	0.14	0.03	0.25	0.33	0.07	0.06	0.09	0.24	0.2	0.1	0.26	0.09	0.13	0.1	0.08	0.03	0.02	0.07	0.04	0.06	0.08	0.07	0.06	0.05	0.03	0.06
B	0.05	0	0.38	0.47	0.09	0.4	0.18	0.33	0.06	0.14	0.29	0.55	0.09	0.07	0.1	0.22	0.23	0.15	0.21	0.06	0.24	0.26	0.15	0.74	0.27	0.44	0.09	0.16	0.11	0.3	0.39	0.77	0.38	0.2	0.09	0.04
C	0.05	0.38	0	0.17	0.09	0.31	0.2	0.15	0.09	0.29	0.31	0.39	0.1	0.08	0.15	0.42	0.36	0.2	0.1	0.04	0.09	0.18	0.22	0.39	0.72	0.42	0.12	0.19	0.26	0.17	0.13	0.3	0.25	0.32	0.21	0.12
D	0.11	0.47	0.17	0	0.07	0.21	0.39	0.31	0.11	0.11	0.77	0.51	0.11	0.19	0.08	0.2	0.1	0.34	0.3	0.06	0.25	0.18	0.25	0.3	0.15	0.26	0.03	0.07	0.06	0.12	0.12	0.18	0.11	0.08	0.04	0.04
E	0.05	0.09	0.09	0.07	0	0.02	0.03	0.07	0.14	0.02	0.03	0.08	0.06	0.06	0.05	0.03	0.05	0.08	0.09	0.57	0.04	0.06	0.06	0.04	0.05	0.04	0.04	0.04	0.05	0.06	0.04	0.03	0.05	0.03	0.02	0.04
F	0.09	0.4	0.31	0.21	0.02	0	0.12	0.31	0.07	0.29	0.17	0.37	0.12	0.09	0.1	0.39	0.14	0.35	0.2	0.03	0.27	0.28	0.31	0.24	0.27	0.18	0.07	0.18	0.4	0.25	0.36	0.24	0.23	0.1	0.06	0.06
G	0.1	0.18	0.2	0.39	0.03	0.12	0	0.07	0.06	0.2	0.29	0.14	0.22	0.12	0.69	0.37	0.17	0.24	0.05	0.05	0.13	0.1	0.29	0.16	0.23	0.28	0.14	0.14	0.06	0.12	0.04	0.07	0.12	0.19	0.17	0.13
H	0.08	0.33	0.15	0.31	0.07	0.31	0.07	0	0.11	0.08	0.1	0.28	0.07	0.12	0.08	0.13	0.07	0.16	0.48	0.05	0.35	0.49	0.12	0.2	0.12	0.11	0.04	0.06	0.14	0.3	0.69	0.25	0.11	0.05	0.04	0.04
I	0.55	0.06	0.09	0.11	0.14	0.07	0.06	0.11	0	0.04	0.05	0.13	0.11	0.22	0.05	0.03	0.04	0.13	0.26	0.15	0.16	0.05	0.06	0.02	0.03	0.07	0.02	0.15	0.05	0.08	0.07	0.05	0.06	0.02	0.03	0.05
J	0.06	0.14	0.29	0.11	0.02	0.29	0.2	0.08	0.04	0	0.25	0.31	0.1	0.03	0.3	0.71	0.37	0.12	0.03	0.07	0.08	0.1	0.17	0.29	0.31	0.26	0.48	0.46	0.28	0.18	0.11	0.11	0.26	0.31	0.28	0.27
K	0.14	0.29	0.31	0.77	0.03	0.17	0.29	0.1	0.05	0.25	0	0.3	0.17	0.14	0.29	0.14	0.27	0.22	0.08	0.08	0.34	0.22	0.31	0.57	0.14	0.2	0.05	0.09	0.12	0.11	0.06	0.15	0.14	0.12	0.06	0.07
L	0.03	0.55	0.39	0.51	0.08	0.37	0.14	0.28	0.13	0.31	0.3	0	0.07	0.05	0.1	0.42	0.31	0.31	0.11	0.06	0.24	0.31	0.19	0.37	0.2	0.55	0.13	0.14	0.15	0.17	0.34	0.28	0.34	0.15	0.07	0.03
M	0.25	0.09	0.1	0.11	0.06	0.12	0.22	0.07	0.11	0.1	0.17	0.07	0	0.61	0.08	0.08	0.11	0.17	0.06	0.08	0.12	0.04	0.18	0.1	0.12	0.1	0.07	0.05	0.06	0.05	0.04	0.07	0.09	0.05	0.13	0.03
N	0.33	0.07	0.08	0.19	0.06	0.09	0.12	0.12	0.22	0.03	0.14	0.05	0.61	0	0.07	0.08	0.06	0.2	0.15	0.08	0.15	0.03	0.09	0.06	0.11	0.07	0.06	0.03	0.03	0.04	0.02	0.04	0.04	0.04	0.11	0.03
O	0.07	0.1	0.15	0.08	0.05	0.1	0.69	0.08	0.05	0.3	0.29	0.1	0.08	0.07	0	0.29	0.29	0.11	0.05	0.05	0.09	0.12	0.26	0.3	0.21	0.21	0.17	0.17	0.06	0.05	0.05	0.13	0.11	0.13	0.15	0.19
P	0.06	0.22	0.42	0.2	0.03	0.39	0.37	0.13	0.03	0.71	0.14	0.42	0.08	0.08	0.29	0	0.47	0.26	0.08	0.08	0.16	0.18	0.2	0.26	0.36	0.45	0.32	0.41	0.31	0.14	0.14	0.17	0.3	0.18	0.28	0.17
Q	0.09	0.23	0.36	0.1	0.05	0.14	0.17	0.07	0.04	0.37	0.27	0.31	0.11	0.06	0.29	0.47	0	0.1	0.03	0.04	0.09	0.12	0.19	0.49	0.51	0.68	0.31	0.3	0.18	0.21	0.08	0.29	0.4	0.52	0.35	0.27
R	0.24	0.15	0.2	0.34	0.08	0.35	0.24	0.16	0.13	0.12	0.22	0.31	0.17	0.2	0.11	0.26	0.1	0	0.17	0.03	0.32	0.2	0.62	0.12	0.09	0.14	0.08	0.08	0.11	0.08	0.08	0.12	0.1	0.06	0.03	0.02
S	0.2	0.21	0.1	0.3	0.09	0.2	0.05	0.48	0.26	0.03	0.08	0.11	0.06	0.15	0.05	0.08	0.03	0.17	0	0.08	0.54	0.18	0.12	0.11	0.07	0.06	0.06	0.04	0.05	0.12	0.28	0.12	0.06	0.06	0.06	0.03
T	0.1	0.06	0.04	0.06	0.57	0.03	0.05	0.05	0.15	0.07	0.08	0.06	0.08	0.08	0.05	0.08	0.04	0.03	0.08	0	0.07	0.06	0.04	0.03	0.03	0.05	0.04	0.04	0.05	0.03	0.03	0.06	0.05	0.05	0.11	0.05
U	0.26	0.24	0.09	0.25	0.04	0.27	0.13	0.35	0.16	0.08	0.34	0.24	0.12	0.15	0.09	0.16	0.09	0.32	0.54	0.07	0	0.45	0.27	0.15	0.08	0.1	0.05	0.06	0.1	0.26	0.12	0.11	0.06	0.07	0.05	0.04
V	0.09	0.26	0.18	0.18	0.06	0.28	0.1	0.49	0.05	0.1	0.22	0.31	0.04	0.03	0.12	0.18	0.12	0.2	0.18	0.06	0.45	0	0.19	0.37	0.24	0.11	0.11	0.12	0.22	0.67	0.45	0.33	0.19	0.09	0.02	0.04
W	0.13	0.15	0.22	0.25	0.06	0.31	0.29	0.12	0.06	0.17	0.31	0.19	0.18	0.09	0.26	0.2	0.19	0.62	0.12	0.04	0.27	0.19	0	0.22	0.23	0.17	0.12	0.17	0.13	0.12	0.06	0.09	0.1	0.06	0.06	0.05
X	0.1	0.74	0.39	0.3	0.04	0.24	0.16	0.2	0.02	0.29	0.57	0.37	0.1	0.06	0.3	0.26	0.49	0.12	0.11	0.03	0.15	0.37	0.22	0	0.46	0.31	0.15	0.19	0.23	0.32	0.31	0.58	0.3	0.2	0.14	0.09
Y	0.08	0.27	0.72	0.15	0.05	0.27	0.23	0.12	0.03	0.31	0.14	0.2	0.12	0.11	0.21	0.36	0.51	0.09	0.07	0.03	0.08	0.24	0.23	0.46	0	0.33	0.23	0.37	0.33	0.2	0.11	0.32	0.26	0.34	0.22	0.16
Z	0.03	0.44	0.42	0.26	0.04	0.18	0.28	0.11	0.07	0.26	0.2	0.55	0.1	0.07	0.21	0.45	0.68	0.14	0.06	0.05	0.1	0.11	0.17	0.31	0.33	0	0.19	0.17	0.2	0.06	0.1	0.32	0.58	0.49	0.2	0.18
N1	0.02	0.09	0.12	0.03	0.04	0.07	0.14	0.04	0.02	0.48	0.05	0.13	0.07	0.06	0.17	0.32	0.31	0.08	0.06	0.04	0.05	0.11	0.12	0.15	0.23	0.19	0	0.63	0.16	0.07	0.12	0.12	0.21	0.37	0.57	0.53
N2	0.07	0.16	0.19	0.07	0.04	0.18	0.14	0.06	0.15	0.46	0.09	0.14	0.05	0.03	0.17	0.41	0.3	0.08	0.04	0.04	0.06	0.12	0.17	0.19	0.37	0.17	0.63	0	0.59	0.23	0.08	0.14	0.25	0.25	0.28	0.19
N3	0.04	0.11	0.26	0.06	0.05	0.4	0.06	0.14	0.05	0.28	0.12	0.15	0.06	0.03	0.06	0.31	0.18	0.11	0.05	0.05	0.1	0.22	0.13	0.23	0.33	0.2	0.16	0.59	0	0.38	0.27	0.34	0.17	0.17	0.09	0.1
N4	0.06	0.3	0.17	0.12	0.06	0.25	0.12	0.3	0.08	0.18	0.11	0.17	0.05	0.04	0.05	0.14	0.21	0.08	0.12	0.03	0.26	0.67	0.12	0.32	0.2	0.06	0.07	0.23	0.38	0	0.56	0.34	0.24	0.13	0.08	0.07
N5	0.08	0.39	0.13	0.12	0.04	0.36	0.04	0.69	0.07	0.11	0.06	0.34	0.04	0.02	0.05	0.14	0.08	0.08	0.28	0.03	0.12	0.45	0.06	0.31	0.11	0.1	0.12	0.08	0.27	0.56	0	0.3	0.18	0.1	0.05	0.05
N6	0.07	0.77	0.3	0.18	0.03	0.24	0.07	0.25	0.05	0.11	0.15	0.28	0.07	0.04	0.13	0.17	0.29	0.12	0.12	0.06	0.11	0.33	0.09	0.58	0.32	0.32	0.12	0.14	0.34	0.34	0.3	0	0.65	0.22	0.08	0.18
N7	0.06	0.38	0.25	0.11	0.05	0.23	0.12	0.11	0.06	0.26	0.14	0.34	0.09	0.04	0.11	0.3	0.4	0.1	0.06	0.05	0.06	0.19	0.1	0.3	0.26	0.58	0.21	0.25	0.17	0.24	0.18	0.65	0	0.65	0.31	0.15
N8	0.05	0.2	0.32	0.08	0.03	0.1	0.19	0.05	0.02	0.31	0.12	0.15	0.05	0.04	0.13	0.18	0.52	0.06	0.06	0.05	0.07	0.09	0.06	0.2	0.34	0.49	0.37	0.25	0.17	0.13	0.1	0.22	0.65	0	0.59	0.39
N9	0.03	0.09	0.21	0.04	0.02	0.06	0.17	0.04	0.03	0.28	0.06	0.07	0.13	0.11	0.15	0.28	0.35	0.03	0.06	0.11	0.05	0.02	0.06	0.14	0.22	0.2	0.57	0.28	0.09	0.08	0.05	0.08	0.31	0.59	0	0.8
N0	0.06	0.04	0.12	0.04	0.04	0.06	0.13	0.04	0.05	0.27	0.07	0.03	0.03	0.03	0.19	0.17	0.27	0.02	0.03	0.05	0.04	0.04	0.05	0.09	0.16	0.18	0.53	0.19	0.1	0.07	0.05	0.18	0.15	0.39	0.8	0

Table A.5: Symmetric Morse-Code Data

A.3 Breakfast Cereal Data

The Breakfast Cereal data used in this dissertation comes from the 1993 Statistical Graphics Exposition organised by the American Statistical Association (Cox and Cox, 2001). The data was made available for any interested party to analyse and present their findings at the exposition. The original data consisted of eleven measurements for seventy-seven different breakfast cereals. For convenience, the subset of the data used by (Cox and Cox, 2001), and this dissertation, consisted of only the 23 cereals (Table A.6) manufactured by the Kellogg brand and ten of the eleven variables. Table A.7 provides the variable information for the data set shown in Table A.8.

Table A.6: Kellogg's Breakfast Cereal

Cereal		**Cereal**	
All Bran	AllB	Just Right Fruit&Nut	JRFN
All Bran: Extra Fiber	AllF	Meuslix Crispy Blend	MuCB
Apple Jacks	AppJ	Nut & Honey Crunch	Nut&
Cornflakes	CorF	Nutri Grain Almond Raisin	NGAR
Corn Pops	CorP	Nutri Grain Wheat	NutW
Cracklin Oat Bran	Crac	Product 19	Prod
Crispix	Cris	Raisin Bran	RaBr
Froot Loops	Froo	Raisin Squares	Rais
Frosted Flakes	FroF	Rice Crispies	RiKr
Frosted Mini Wheats	FrMW	Smacks	Smac
Fruitful Bran	FruB	Special K	Spec
Just Right Crunch Nuggets	JRCN		

Table A.7: Breakfast Cereal Variables

Variable	
Number of Calories	Cal
Protein(g)	Prot
Fat (g)	Fat
Sodium (mg)	Na
Dietary Fiber (g)	DF
Complex Carbohydrates (g)	CC
Sugars (g)	Sug
Display Shelf	Shelf
Potassium (mg)	K
Vitamins & Minerals	V&M

Table A.8: Kellog's Cereal Data

	Cal	Prot	Fat	Na	DF	CC	Sug	Shelf	K	V&M
AllB	70	4	1	260	9	7	5	3	320	25
AllF	50	4	0	140	14	8	0	3	330	25
AppJ	110	2	0	125	1	11	14	2	30	25
CorF	100	2	0	290	1	21	2	1	35	25
CorP	110	1	0	90	1	13	12	2	20	25
Crac	110	3	3	140	4	10	7	3	160	25
Cris	110	2	0	220	1	21	3	3	30	25
Froo	110	2	1	125	1	11	13	2	30	25
FroF	110	1	0	200	1	14	11	1	25	25
FrMW	100	3	0	0	3	14	7	2	100	25
FruB	120	3	0	240	5	14	12	3	190	25
JRCN	110	2	1	170	1	17	6	3	60	100
JRFN	140	3	1	170	2	20	9	3	95	100
MuCB	160	3	2	150	3	17	13	3	160	25
Nut	120	2	1	190	0	15	9	2	40	25
NGAR	140	3	2	220	3	21	7	3	130	25
NutW	90	3	0	170	3	18	2	3	90	25
Prod	100	3	0	320	1	20	3	3	45	100
RaBr	120	3	1	210	5	14	12	2	240	25
Rais	90	2	0	0	2	15	6	3	110	25
RiKr	110	2	0	290	0	22	3	1	35	25
Smac	110	2	1	70	1	9	15	2	40	25
Spec	110	6	0	230	1	16	3	1	55	25

A.4 SynTReN EColi Microarray Data

The SynTReN (Synthetic Transciptional Regulatory Networks) *Java* software package is a useful tool for generating, microarray like, gene expression data. The data set here is generated from an Ecoli source network which comes as a default standard for the SynTReN package. The generated data was set to have fifty genes and one hundred samples, with the default values for all noise parameters. In microarray experiments, genes are the subjects and samples are the variables. Therefore, the data is 50×100 in size. Due to the magnitude of this data set, only the names of the 50 genes are included here and can be found in Table A.9.

Table A.9: Microarray Ecoli Genes

lon	rpoH	clpP	dnaKJ
grpE	hflB	htpG	htpY
ibpAB	mopA	mopB	dnaA
nrdAB	rpoE_rseABC	ecfI	htrA
skp_lpxDA_fabZ	xprB_dsbC_recJ	cutC	dapA_nlpB_purA
ecfABC	ecfD	ecfF	ecfG
ecfH	ecfJ	ecfK	ecfLM
fkpA	ksgA_epaG_epaH	lpxDA_fabZ	mdoGH
nlpB_purA	ostA_surA_pdxA	rfaDFCL	rpoD
uppS_cdsA_ecfE	yhdG_fis	sdhCDAB_b0725_sucABCD	aldB
proP	adhE	pdhR_aceEF_lpdA	alaWX
argU	argW	argX_hisR_leuT_proM	aspV
leuQPV	leuX		

Appendix B

Supporting Documentation

The three documents provided in this Appendix Chapter have been written to accompany the MDS-GUI and will all be available to the public when downloading the software. Each one is a stand alone document. The three documents are: the Reference Manual, the User Manual and the package Vignette. A description of each will be given before the document itself which will inform on its purpose. It should be noted that each document has a Bibliography included in their stand alone versions, but these have been omitted from this complete dissertation.

B.1 CRAN Package Reference Manual

The following pages show the reference manual required to be provided as part of the submission of an *R* package to the CRAN database. The general outline of the required document is that it first provides details of the package and its developers and then goes on to describe all functions and datasets found within the package. The **MDSGUI** package has one sole function, being the `MDSGUI` function and so only gives information on this function. The package holds no data sets since data is uploaded to the GUI from outside the *R*-Environment. The format requirements of the reference manual are very strict and were not subject to the author's preferences.

Package ‘MDSGUI’

July 13, 2012

Type Package

Title A GUI for interactive MDS in R.

Version Version 0.1

Date 2012-07-13

Author Andrew Timm, Sugnet Lubbe and code contribution by NJ le Roux.

Maintainer Andrew Timm <timmand@gmail.com>

Depends R (2.13.0 preferable) otherwise (≥=2.15.1), rgl(≥=0.92), tcltk2 (1.1-5 preferable), tkrplot (≥= 0.0-23).

Imports tkrplot, tcltk2, tcltk, MASS, boot, RColorBrewer, rgl and scatterplot3d.

Description A graphical user interface (GUI) for performing Multidimensional Scaling applications and interactively analysing the results all within the GUI environment. The MDS-GUI provides means of performing Classical Scaling, Least Squares Scaling, Metric SMACOF, Non-Metric SMACOF, Kruskal’s Analysis and Sammon Mapping with animated optimisation.

License GPL

OS-type Windows

LazyLoad yes

Repository CRAN

Date/Publication ...

R topics documented:

MDSGUI 3

Index 4

`MDSGUI` *A GUI for interactive MDS in R*

Description

A graphical user interface (GUI) for performing Multidimensional Scaling applications and interactively analysing the results all within the GUI environment. The MDS-GUI provides means of performing Classical Scaling, Least Squares Scaling, Metric SMACOF, Non-Metric SMACOF, Kruskal's Analysis and Sammon Mapping with animated optimisation.

Usage

```
MDSGUI()
```

Details

`MDSGUI` is the sole function of the **MDSGUI** package. Data used in the GUI is not input as an argument of the function like some other R GUIs. Instead the data is uploaded directly into the GUI when it is running by means of the native Windows load file window. All functions, features and parameter adjustments are done within the MDS-GUI itself and therefore no coding is required by the user apart from creating an instance of it. The software features and menu navigation manual is accessed via the *Help* menu in the MDS-GUI. A vignette demonstrating its use may also be found there. At present, **MDSGUI** is only available on the Windows operating system.

Author(s)

Andrew Timm `<timmand@gmail.com>` & Sugnet Lubbe

References

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

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

Examples

```
## Not run:
MDSGUI()
## End(Not run)
```

Index

∗Topic **dynamic**
MDSGUI, 3

∗Topic **multivariate**
MDSGUI, 3

B.2 MDS-GUI User Manual

The User Manual for the MDS-GUI, which will be available directly through the GUI via the *User Manual* option of the *Help* menu, is provided here. This manual describes each of the components of the program from a software description point of view. No statistics are discussed in this manual, but rather each of the menus and areas are described. The document was designed with the use of hypertext links which provide direct links to the sections of the document when referenced in other locations. The document is best viewed electronically as with most online based documentation. The first pages of the user manual provide further description and explanation of the format and layout of the document.

UNIVERSITY OF CAPE TOWN

DEPARTMENT OF STATISTICAL SCIENCES

MDS-GUI

Users Manual

Andrew Timm and Sugnet Lubbe

The MDS-GUI Version 0.1

Draft Version

August 8, 2012

Contents

The MDS-GUI Manual 5
GUI Areas 6
- Main Plotting Area 6
- Plotting Tabs 7
- Secondary Plotting Area 7
- Table Section 7
- Information Panel 7

Available Plots 7
- MDS Configuration Plot 7
- Shepard Plot 7
- Stress Plot 8
- Log Stress 8
- Scree Plot 8
- Zoomed Plot 8
- RGL 3D Plot 8
- Static 3D Plot 8
- Procrustes Analysis Plot 8
- Popped-Out Plot 8

Top-Menu 8
- File 9
  - New User 9
  - Save User Workspace 9
  - Load User Workspace 9
  - Print 9
  - Clear All 9
  - Exit MDS-GUI 9
- General 9
  - Undo 9
  - Appearance Settings 9
  - General Settings 10
  - Export 10
    - All 9
    - Plot (1-5) info 9
- Data 10
  - Load Dataset 11
  - Load Dissimilarity Matrix 11
  - Load Similarity Matrix 11
  - Correlation Matrix 12
  - Data Colour Index 12
  - Colour Categories 12
  - Edit 12
    - Active Data 12
    - Active Dissimilarity Matrix 12
    - Active Similarity Matrix 13
    - Active Correlation Matrix 13
    - Active Coordinate Vectors 13
  - Save 13
    - Dataset 13
    - Dissimilarity Matrix 13
    - Similarity Matrix 13
    - Correlation Matrix 13
    - MDS Coordinate Matrix 13
  - Data Options 13
- Multivariate Tools 13
  - MDS 13
    - Classical Scaling 13
    - Metric Symmetric SMACOF 13
    - Least Squares Scaling 14
    - Non-Metric Symmetric SMACOF 14
    - Kruskal's Analysis 14
    - Sammon Mapping 14
  - MDS Options 14

Procrustes Analysis 14
Dissimilarity Matrix Calculation 15
Euclidean 15
Weighted Euclidean 15
Mahalanobis Distance 15
City Block Metric 15
Minkowski Metric 15
Canberra Metric 15
Divergence 16
Soergel 16
Bhattacharyya Distance 16
Wave-Hedges 16
Angular Separation 16
Correlation 16
Help 16
Function Code 16
Display Pop-Out Code 17
Vignette 17
User Manual 17
Homepage 17
About 17
Main Plot Menu 18
Label Specific Point 18
Clear Added Point Labels 18
Relocate a Group of Points 18
Remove a Point 18
Use Coordinates as Starting Configuration . . 18
Rotate and Reflect 19
Advanced Zoom 19
Change Point Colour 20
Default Point Colours 21
Display Variable Axes 21
Remove Axes of Variable(s) 21
Pop-out Plot 21
Copy Plot to Clipboard 21
Plot Options 21
Secondary Plot Menu 22
Clear Added Labels 22
Label Specific Point 22
Pop-Out Enlarged Plot 22
Copy (Plot name) to Clipboard 23
(Plot Name) Plot Options 23
General Settings 23
General Tab 23
Convergence Tab 24
Graphical Tab 25
Visualisation Tab 26
Data Options Menu 27
MDS Options Menu 28
Dimensions Tab 28
Starting Configuration Tab 28
Stress Tab 29
Plot Options Menu 30
General Tab 30
Points Tab 31
Lines Tab 32
Axes Tab 33
Other Features 34
NotesScript 34

Run as Code 34
Save Notes 34
Load Notes 34
Information Table 34
MDS Configurations 34
Removed Points 35
Removed Axes 35
Animated Optimisation 35
End Process 35
Keyboard Shortcuts 35
Known Issues 36

List of Figures

Figure 1: The MDS-GUI 6
Figure 2: New User Window 9
Figure 3: Appearance Settings 10
Figure 4: Export Instructions 10
Figure 5: New Active Dataset 11
Figure 6: Colour Categories Options 12
Figure 7: Procrustes Options 14
Figure 8: Function Help Menu 17
Figure 9: Label Specific Point 18
Figure 10: Rotate and Reflect 19
Figure 11: Advanced Zoom 20
Figure 12: Variable Axes Removal Options 21
Figure 13: Shepard Point Label 22
Figure 14: General Tab of General Settings 23
Figure 15: Convergence Tab of General Settings 24
Figure 16: Graphical Tab of General Settings 25
Figure 17: Visualisation Tab of General Settings 26
Figure 18: Data Options Menu 27
Figure 19: Dimensions Tab 28
Figure 20: Starting Configurations Tab 29
Figure 21: Stress Tab 30
Figure 22: Plot Options: General Tab 31
Figure 23: Plot Options: Points Tab 32
Figure 24: Plot Options: Lines Tab 33
Figure 25: Plot Options: Axes Tab 33

The MDS-GUI Users Manual

This manual serves to provide information regarding the layout and features of the MDS-GUI (Multidimensional Scaling Graphical User Interface). The four sections of this document will cover: the layout of the GUI, the various plots, the menus and features of the GUI, and finally features of the GUI that are not menu based.

The layout of the document is point form in an indented four tier format. In descending order, the tiers are represented by the following symbols: $\bullet$, $\diamond$, $*$, $\circ$. The manual will make use of screenshots of GUI in the descriptions of menus and features, each of which will be described in the text. The document will also make extensive use of hypertext links for those reading the manual electronically. To avoid confusion, when menus are discussed, this will be indicated by '`MENU`' at the beginning of the paragraph. Discussion of features will be indicated by 'FEATURE'. In addition, when a pop-out options window is discussed, the options in which it contains will be presented in a framed text box.

The users manual will not contain any theory on Multidimensional Scaling or any other Multivariate techniques. For a useful source on MDS techniques, read *Multidimensional Scaling: Second Edition* (Cox and Cox, 2001) and *Modern Multidimensional Scaling Theory and Applications: Second Edition* (Borg and Groenen, 2005). For more information on the practical use of the MDS-GUI and MDS result interpretation, see the Vignette of the **MDSGUI** package.

The MDS-GUI was developed with and makes use of a number of other *R* packages. These include: **tkrplot** (Tierney, 2011), **tcltk2** (Grosjean, 2011), **MASS** (Venables and Ripley, 2002), **boot** (Canty and Ripley, 2010), **RColorBrewer** (Neuwirth, 2011), **rgl** (Adler and Murdoch, 2011) and **scatterplot3d** (Ligges and Mächler, 2003).

GUI Areas

The MDS-GUI has a layout with various sections. Figure 1: The MDS-GUI below shows the default view of the GUI with each of its major sections labeled one to five. A description of each of these areas will now be given in general terms.

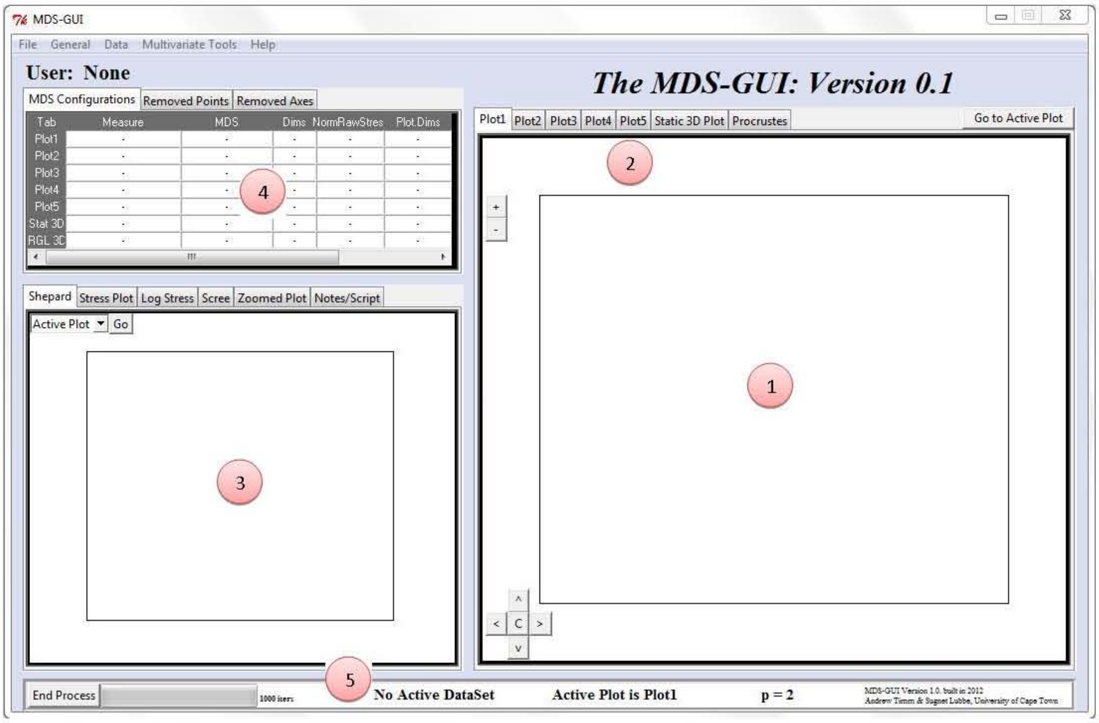


Figure 1: The MDS-GUI

Main Plotting Area 1: The area labeled as 'one' is the area on which the • MDS Configuration Plot in two dimensions ($p = 2$ is default). The Euclidean distances, d, are illustrated in area one. The aspect ratio of the plotting area is one thus preserving the interpretation of the distances regardless of the orientation of the axes.

Plotting Tabs 2: Five plotting tabs are available to perform independent MDS procedures with individual settings. These results may then be compared. In addition, the 'Static 3D Plot' tab shows • Static 3D Plot, the output of the two dimensional depiction of the result when $p = 3$ and the 'Procrustes' tab shows the result of a Procrustes analysis between two separate configurations with the • Procrustes Analysis Plot.

Secondary Plotting Area 3: This smaller area houses multiple diagnostic outputs generated as a default for most of the MDS processes. Each of the Tabs of the area show a different utility. These include: • Shepard Plot, • Stress Plot, • Log Stress, • Scree Plot, • Zoomed Plot and • NotesScript.

Table Section 4: The window on the top left of the GUI holds the relevant tables included in the software. The front most table is called the ◇ MDS Configurations table and holds important information relating to each of the MDS configurations in all main plotting tabs. The second and third tables are called the ◇ Removed Points and ◇ Removed Axes tables respectively. Features of the MDS-GUI include the option to ◇ Remove a Point from the configuration and also ◇ Remove Axes of Variable(s) from the display. The information contained in these tables pertain to these scenarios.

Information Panel 5: The final numbered label refers to the Information Pane located at the bottom of the GUI. This area displays various information relevant to the applications of the user. The pane includes information regarding the data set being used, the current plotting area and the software developer details.

Available Plots

• **MDS Configuration Plot** : When $p = 2$, this plot is found in the Main Plotting Area.

`FEATURE` – The MDS Configuration Plot displays the $\mathbf{X}{:}n \times p$ configuration from the MDS process. When $p = 3$ it can be shown in the • Static 3D Plot.

`FEATURE` – Clicking any point on the configuration with the mouse will label that point.

`FEATURE` – May Zoom in and Out manually using the '+' and '-' buttons next to the plot.

`FEATURE` – May move the configuration left, right, up, down, or return to original orientation using the $\leftarrow$, $\rightarrow$, $\uparrow$, $\downarrow$ and 'C' buttons next to the plot respectively.

`FEATURE` – May drag individual points manually around the plot by clicking and holding the left mouse button. This causes real time changes to both • MDS Configuration Plot, • Shepard Plot and stress value shown in the ◇ MDS Configurations tab of the • Information Table.

`FEATURE` – The • Main Plot Menu provides numerous other features of this plot.

• **Shepard Plot** : Found in the Shepard tab of Secondary Plotting Area.

`FEATURE` – Plots d_{ij} vs. δ_{ij}.

`FEATURE` – Clicking any point on the plot will highlight the point and draw the connection between the corresponding point on the • MDS Configuration Plot. Both point and line will be the same colour. The first ten selected points will be labeled and thereon none are labeled.

`FEATURE` – 'Brushing' the plot by dragging a box over the plot will label each of the points contained in the box as described above.

FEATURE – The appropriate • Secondary Plot Menu provides further features for this plot.

- **Stress Plot** : Found in the Stress Plot1 tab of Secondary Plotting Area.

  FEATURE – Plots stress vs. iterations.

  FEATURE – The appropriate • Secondary Plot Menu provides further features for this plot.

- **Log Stress** : Found in the Stress Plot2 tab of Secondary Plotting Area.

  FEATURE – Plots the logged difference of stress values over iterations

  FEATURE – The appropriate • Secondary Plot Menu provides further features for this plot.

- **Scree Plot** : Found in the Scree tab of Secondary Plotting Area.

  FEATURE – Plots stress over dimensions.

  FEATURE – Illustrates both current dimension and optimum dimension.

  FEATURE – The appropriate • Secondary Plot Menu provides further features for this plot.

- **Zoomed Plot** : Found in the Zoomed Plot tab of Secondary Plotting Area.

  FEATURE – Shows an isolated zoomed area of Main Plotting Area when ⋄ Advanced Zoom is used.

- **RGL 3D Plot** : Plot is placed in the *R* console and uses the rgl package (Adler and Murdoch, 2011).

  FEATURE – Produces dynamic 3D rendition of the configuration when $p = 3$.

  FEATURE – User has full control over plot in terms of rotation and zoom.

- **Static 3D Plot** :Plot is placed in the Static 3D Plot tab of Plotting Tabs. Uses scatterplot3d package (Ligges and Mächler, 2003).

  FEATURE– Produces static 3D rendition of the configuration when $p = 3$.

  FEATURE– User May rotate the horizontal plane of the plot using the $\leftarrow$ and $\rightarrow$ buttons.

  FEATURE– User may extend the length of the horizontal plane using the $\uparrow$ and $\downarrow$ buttons. 'C' buttons returns to original position.

  FEATURE– The appropriate • Secondary Plot Menu provides further features for this plot.

- **Procrustes Analysis Plot** : Plot is placed in the Procrustes tab of Plotting Tabs.

  FEATURE – Result of Procrustes Analysis from any two separate active plots in Plotting Tabs.

  FEATURE– The appropriate • Secondary Plot Menu provides further features for this plot.

- **Popped-Out Plot** :

  FEATURE – All of the above plots may be popped out to a separate window from their respective right click menus.

Menus → Functions

- **Top-Menu** : The menu system found across the top panel of the GUI.

  MENU– The top menu provides access to the majority of processes contained in the MDS-GUI.

◇ **File** : The first menu option of • Top-Menu.

MENU- The file menu focuses on the handling of the specific files and workspaces of the user.

∗ **New User** : Opens the New User Window.

MENU- Provides means of the user inputting their name for convenient research labeling.

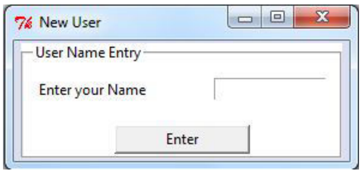


Figure 2: New User Window

Options of New User Window

Enter Your Name : Requires Text Input

FEATURE- Provided name will reflect on the MDS-GUI frontend and the PDF files when ∗ Export is used.

∗ **Save User Workspace** : Opens native operating system Save-File window

FEATURE- Saves all plots, figures and settings to external file of user's choosing.

∗ **Load User Workspace** : Opens native operating system Load-File window

FEATURE- Load all plots, figures and settings from external file of user's choosing. Replots automatically.

∗ **Print** : Opens native operating system print options window.

FEATURE- Prints the • MDS Configuration Plot of the topmost plotting tab of Plotting Tabs

∗ **Clear All** : Provides fresh workspace

FEATURE- Clears all plots, details and restores setting defaults.

∗ **Exit MDS-GUI** : Opens window asking user whether they would like to save their workspace before quitting.

FEATURE- Exits workspace safely after either saving or not saving the workspace.

◇ **General** : The second menu option of • Top-Menu.

MENU- Contains wide range of options concerning the GUI and its internal settings.

∗ **Undo** : Available only after the • MDS Configuration Plot has been manually altered in some way. Either by dragging or ◇ Relocate a Group of Points.

FEATURE- The most recent alteration made to the configuration is reverted back to its previous state. Changing plot, performing another MDS procedure or making another alteration removes opportunity to undo the original alteration.

∗ **Appearance Settings** : Opens the Appearance Settings window.

MENU- Provides user with means to make colour adjustments to the MDS-GUI itself.

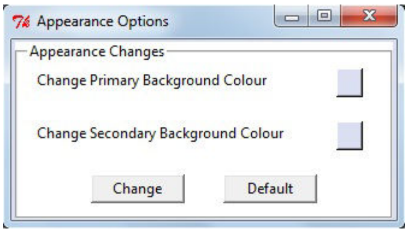


Figure 3: Appearance Settings

Options of Appearance Settings

Change Primary Background Colour : Click Colour Box to open native colour choice window.

FEATURE- Select colour for background of the MDS-GUI.

Change Secondary Background Colour : Click Colour Box to open native colour choice window.

FEATURE- Select colour for background of pop out plots and windows.

Change and Default : Buttons to either make selection or revert to defaults.

∗ **General Settings** : Opens the ● General Settings menu.

∗ **Export** : Uses Sweave (Leisch, 2002) and latex to create a PDF file. Any selection produces the important instructions shown in Export Instructions

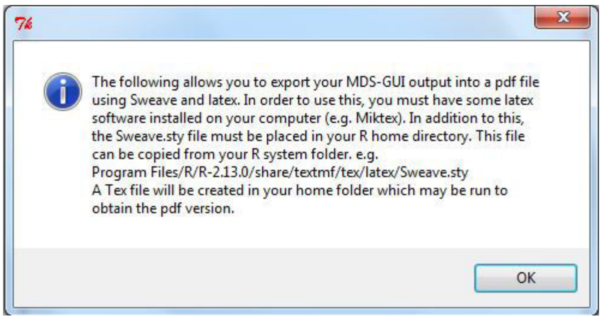


Figure 4: Export Instructions

FEATURE- In all cases, the PDF produced will provide all stress values, MDS configuration, Shepard Plot and Scree Plot for every included plotting area.

∘ **All** : Produces a PDF document with information for all utilised plotting areas.

∘ **Plot (1-5) info** : Produces PDF document for specific plotting area.

⋄ **Data** : The third menu option of ● Top-Menu.

MENU- Provides all options relating to the user upload of data on which the MDS procedures are performed. The data may originally be provided in a number of different forms, all of which are allowable by the MDS-GUI. The user is able to upload: a samples by variables matrix $\mathbf{Z}$:$n \times m$;

a distance matrix Δ:$n \times n$; or an $n \times n$ similarity or correlation matrix. In the case where Δ is not uploaded, appropriate calculations and transformations are performed automatically in order to construct Δ, which is used in all MDS procedures.

$*$ **Load Dataset** : Opens the New Active Dataset window.

FEATURE-Uploads $\mathbf{Z}$:$n \times m$ matrix. Δ is calculated from this using the selected option from $*$ Dissimilarity Matrix Calculation.

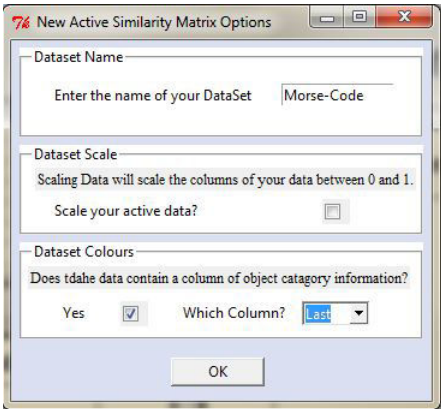


Figure 5: New Active Dataset

Options of New Active Dataset

Enter the name of your Dataset : Requires Text input

FEATURE- The name of the data will appear on Information Panel, all of the resulting MDS plots and $*$ Export documents.

Transpose Active Data : Requires checkbox selection

FEATURE- Δ is created exclusively from a matrix with objects as rows and variables as columns. If this is the other way round, the dissimilarity matrix will be m$\times$m and treat variables as objects. Selecting this allows for a correction if the data has variables $\times$ objects.

Scale your Active Data : Require checkbox selection

FEATURE- Provides option to scale each variable column to range between 0 and 1.

Category Information : Only applicable when a column of categories is present. Requires checkbox selection and indication of where category column is in data.

FEATURE- This column is stored only as categorical information and removed from the data. All objects are assigned a colour according to their defined category.

OK : Executes all specified assignments on data.

$*$ **Load Dissimilarity Matrix** : Produces window similar to New Active Dataset.

FEATURE- Uploads Δ directly.

$*$ **Load Similarity Matrix** : Produces window similar to New Active Dataset.

FEATURE- Uploads $\mathbf{S}$ matrix. Δ is calculated by scaling $\mathbf{S}$ and subtracting it from the $n \times n$ $\mathbf{1}$

matrix.

∗ **Load Correlation Matrix** : Produces window similar to New Active Dataset.

`FEATURE-` Uploads **R** matrix. Δ is calculated by $n \times n$ **1** matrix - **R**.

∗ **Data Colour Index** : Produces a matrix editor window using a variation of the **tcltk2** (Grosjean, 2011) `tk2edit` function.

`FEATURE-`Each object is displayed with their corresponding colour code. Manual alteration of this to another legitimate code will change the points display on the • MDS Configuration Plot.

∗ **Colour Categories** : Opens the Colour Categories Options window.

`MENU-` Provides means of making visual adjustments to the way the defined categories are shown on the • MDS Configuration Plot

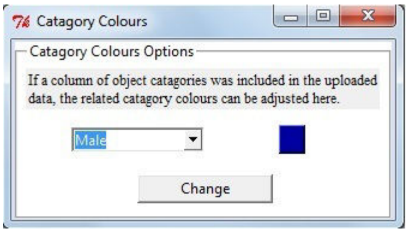


Figure 6: Colour Categories Options

Options of Colour Categories Options

Category Selection : Use drop down box to select category.

`FEATURE-`Each category option listed when uploading data will be present in this list.

Colour Box : Click box to bring up native operating system colour selection window.

`FEATURE-` Whichever colour is selected will be reflected immediately both in the • MDS Configuration Plot and the ∗ Data Colour Index. This setting holds for all plotting areas.

Change : Select button to make all appropriate changes.

∗ **Edit** Opens the Edit submenu.

`MENU-` Provides user access to view and make alterations to every applicable form of data in the MDS-GUI instance.

◦ **Active Data** : Produces a matrix editor window using a variation of the **tcltk2** (Grosjean, 2011) `tk2edit` function.

`FEATURE-` Displays the **Z**:$n \times m$ data matrix (if data uploaded using ∗ Load Dataset option). Elements of the matrix can be altered. Valid changes are reflected with immediate effect.

◦ **Active Dissimilarity Matrix** : Produces a matrix editor window using a variation of the **tcltk2** (Grosjean, 2011) `tk2edit` function.

`FEATURE-` Displays the Δ:$n \times n$ dissimilarity matrix of the active plotting area of the Plotting Tabs. Elements of the matrix can be altered. Valid changes are reflected with immediate effect.

◦ **Active Similarity Matrix** : Procudes a matrix editor window using a variation of the **tcltk2** (Grosjean, 2011) `tk2edit` function.

`FEATURE-` Displays the **S**:$n \times n$ matrix (if data uploaded using the ∗ Load Similarity Matrix option). Valid changes are reflected with immediate effect.

◦ **Active Correlation Matrix** : Produces a matrix editor window using a variation of the **tcltk2** (Grosjean, 2011) `tk2edit` function.

`FEATURE-` Displays the correlation **R**:$n \times n$ matrix (if data uploaded using the ∗ Load Correlation Matrix option). Valid changes are reflected with immediate effect.

◦ **Active Coordinate Vectors** : Produces a matrix editor window using a variation of the **tcltk2** (Grosjean, 2011) `tk2edit` function.

`FEATURE-` Displays the **X**:$n \times p$ coordinate matrix of the active plotting area of the Plotting Tabs. Valid changes are reflected with immediate effect.

∗ **Save** : Opens the Save submenu.

`MENU-` Provides user access to save the relevant datasets to external files.

◦ **Dataset** : Opens the native operating system save-file window.

`FEATURE-` The **Z**:$n \times m$ data matrix (if data uploaded using ∗ Load Dataset option) is saved to an external file for external use.

◦ **Dissimilarity Matrix** : Opens the native operating system save-file window.

`FEATURE-` The active Δ dissimilarity matrix is saved to an external file for external use.

◦ **Similarity Matrix** : Opens the native operating system save-file window.

`FEATURE-` The **S**:$n \times n$ similarity matrix (if data uploaded using ∗ Load Similarity Matrix option) is saved to an external file for external use.

◦ **Correlation Matrix** : Opens the native operating system save-file window.

`FEATURE-` The correlation **R**:$n \times n$ similarity matrix (if data uploaded using ∗ Load Correlation Matrix option) is saved to an external file for external use.

◦ **MDS Coordinate Matrix** : Opens the native operating system save-file window.

`FEATURE-` The active **X**:$n \times p$ is saved to an external file for external use.

∗ **Data Options** : Opens the • Data Options Menu options window.

⋄ **Multivariate Tools** : Opens the fourth menu option of • Top-Menu.

`MENU-` This menu provides access to all the multivariate capabilities of the MDS-GUI.

∗ **MDS** : Opens the MDS submenu.

`MENU-`The MDS menu allows for use of the six Multidimensional Scaling methods utilisable in the MDS-GUI.

◦ **Classical Scaling** : Uses `cmdscale` function from **stats** package (R Development Core Team, 2012).

`FEATURE-` Performs the metric method called Classical Scaling on Δ.

◦ **Metric Symmetric SMACOF** : Uses adaptation of contributed code by le Roux (2012).

`FEATURE-`Performs the metric method called Metric SMACOF on Δ.

◦ **Least Squares Scaling** : Uses original code.

`FEATURE`- Performs the metric method called Least Squares Scaling on Δ.

◦ **Non-Metric Symmetric SMACOF** : Uses adaptation of contributed code by le Roux (2012).

`FEATURE`-Performs the non-metric method called Non-Metric SMACOF on Δ.

◦ **Kruskal's Analysis** : Uses `isoMDS` function from **MASS** package (Venables and Ripley, 2002).

`FEATURE`-Performs the non-metric method called Kruskal's method on Δ.

◦ **Sammon Mapping** : Uses `sammon` function from **MASS** package (Venables and Ripley, 2002).

`FEATURE`-Performs the non-metric method called Sammon Mapping on Δ.

∗ **MDS Options** : Opens the • MDS Options Menu window.

∗ **Procrustes Analysis** : Produces Procrustes Options window.

`MENU`- Gives user control over the specifics of their desired Procrustes Analysis.

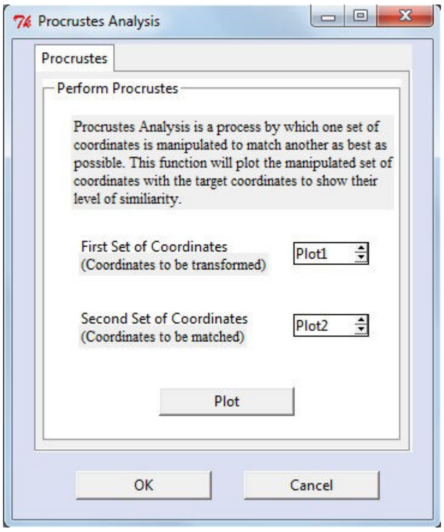


Figure 7: Procrustes Options

Options of Procrustes Options

First Set of Coordinates : Requires selection of applicable plot from drop down list.

`FEATURE`- The selection will be the coordinates that are transformed.

Second Set of Coordinates : Requires selection of applicable plot from drop down list.

`FEATURE`- The selection will be the coordinates that are to be matched.

Plot : Selection Button.

`FEATURE`- When both selections are valid and contain equivalent configurations, this selection will display the • Procrustes Analysis Plot in the Procrustes tab of Plotting Tabs.

Cancel : Selection Button.

`FEATURE-` Exits the menu with no selections being made.

∗ **Dissimilarity Matrix Calculation** : Opens the Dissimilarity Matrix Calculation submenu.

`MENU-`The Dissimilarity Matrix Calculation menu is in radiobutton format. That is only one of the menu options may be selected at a time and the active selection is indicated by a tick mark on the menu. The menu is only available when data is loaded through ∗ Load Dissimilarity Matrix.

◦ **Euclidean** : Radiobutton menu selection option.

`FEATURE-` Δ calculated using the Euclidean Metric.

$$\delta_{rs} = \sqrt{\sum_i (x_{ri} - x_{si})^2}$$

◦ **Weighted Euclidean** : Radiobutton menu selection option.

`FEATURE-` Δ calculated using the Weighted Euclidean Metric.

$$\delta_{rs} = \sqrt{\sum_i w_i (x_{ri} - x_{si})^2}$$

◦ **Mahalanobis Distance** : Radiobutton menu selection option.

`FEATURE-` Δ calculated using the Mahalanobis Distance Metric.

$$\delta_{rs} = \sqrt{(\mathbf{x}_r - \mathbf{x}_s)^T \Sigma^{-1} (\mathbf{x}_r - \mathbf{x}_s)}$$

◦ **City-Block Metric** : Radiobutton menu selection option.

`FEATURE-` Δ calculated using the City-Block Metric.

$$\delta_{rs} = \sum_i |x_{ri} - x_{si}|$$

◦ **Minkowski Metric** : Radiobutton menu selection option.

`FEATURE-` Δ calculated using the Minkowski Metric.

$$\delta_{rs} = \sqrt[\lambda]{\sum_i w_i |x_{ri} - x_{si}|^\lambda}$$

◦ **Canberra Metric** : Radiobutton menu selection option.

`FEATURE-` Δ calculated using the Canberra Metric.

$$\delta_{rs} = \sum_i \frac{|x_{ri} - x_{si}|}{(x_{ri} + x_{si})}$$

○ **Divergence** : Radiobutton menu selection option. Not available when the uploaded **Z** matrix contains zeros.

`FEATURE-` Δ calculated using the Divergence Metric.

$$\delta_{rs} = \frac{1}{p}\sum_i \frac{(x_{ri}-x_{si})^2}{(x_{ri}+x_{si})^2}$$

○ **Soergel** : Radiobutton menu selection option.

`FEATURE-` Δ calculated using the Soergel Metric.

$$\delta_{rs} = \frac{\sum_i |x_{ri}-x_{si}|}{\sum_i max(x_{ri},x_{si})}$$

○ **Bhattacharyya Distance** : Radiobutton menu selection option.

`FEATURE-` Δ calculated using the Bhattacharyya Metric.

$$\delta_{rs} = \sqrt{\sum_i (x_r^{\frac{1}{2}}{}_i - x_s^{\frac{1}{2}}{}_i)}$$

○ **Wave-Hedges** : Radiobutton menu selection option. Not available when the uploaded **Z** matrix contains zeros.

`FEATURE-` Δ calculated using the Wave-Hedges Metric.

$$\delta_{rs} = \frac{1}{p}\sum_i \left(1 - \frac{min(x_{ri},x_{si})}{max(x_{ri},x_{si})}\right)$$

○ **Angular Separation** : Radiobutton menu selection option.

`FEATURE-` Δ calculated using the Angular Separation Metric.

$$\delta_{rs} = 1 - \frac{\sum_i x_{ri}x_{si}}{\sqrt{[\sum_i x_{ri}^2 \sum_i x_{si}^2]}}$$

○ **Correlation** : Radiobutton menu selection option.

`FEATURE-` Δ calculated using Correlation.

$$Pearson's : s_{rs} = \frac{\sum_i (x_{ri}-\bar{x}_r)(x_{si}-\bar{x}_s)}{\sqrt{\sum_i (x_{ri}-\bar{x}_r)^2 \sum_i (x_{si}-\bar{x}_s)^2}}$$

◇ **Help** : Opens the fifth menu of the • Top-Menu.

`MENU-` Provides the user with assistance and details regarding the MDS-GUI itself.

∗ **Function Code** : Opens the Function Help Menu Window.

`MENU-` Gives options relating to the code behind the MDS-GUI.

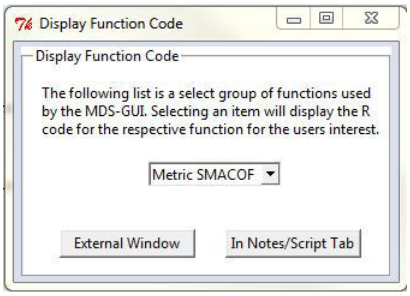


Figure 8: Function Help Menu

Options for Function Help Menu

Choose Function : Drop-down menu selection.

FEATURE- Whichever function name is selected from the list will have its code displayed for the users convenience in the environment that they choose. This is purely for observational purposes and any 'changes' made to the code will not have any effect to the MDS-GUI. User are however able to copy the code in a script file to easily make their own alterations.

External Window : Section Button.

FEATURE- Selection of this option will display the chosen function's code in an external tcltk popped out text box.

In NotesScript Tab : Section Button.

FEATURE- Selection of this option will display the chosen functions's code in the • NotesScript tab in the Secondary Plotting Area.

* **Display Pop-Out Code** : Acts as a checkbox in the menu. Can be activated or deactivated via the menu with the result reflected by a checkmark.

  FEATURE- When active, hovering the mouse cursor over various areas of the GUI will produce pop up text boxes that provide help information.

* **Vignette** : Internet Link

  FEATURE-Opens the vignette PDF document. Internet connection required.

* **User Manual** : Internet Link

  FEATURE-Opens this PDF document. Internet connection required.

* **Homepage** : Internet Link

  FEATURE-Directs user to both the website for the Department of Statistical Sciences for the University of Cape Town and the CRAN page for the **MDSGUI** package.

* **About** : Opens Information Text Box.

  FEATURE- Provides information about the software and developers.

• **Main Plot Menu** : Menu is accessed via right clicking whilst the mouse is over the Main Plotting Area.

MENU- The menu called is specific to the plotting tab selected in the Plotting Tabs area. The changes and applications performed from the menu are localised to the active tab from which the menu was called. The tab to which the menu is connected with is shown at the top of the menu.

◇ **Label Specific Point** : Opens the Label Specific Point window.

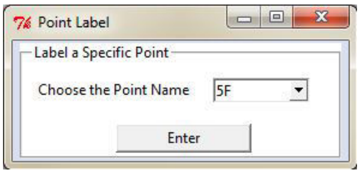


Figure 9: Label Specific Point

Options for Label Specific Point

Choose the Point Name : Selection is required from drop down list of object names.

FEATURE- This feature is useful when there are many points on the • MDS Configuration Plot and a specific object needs to be located. Selection of the object will label the point appropriately.

Enter : Selection Button.

FEATURE- When pressed the selected point will be labeled.

◇ **Clear Added Point Labels** : Alters the • MDS Configuration Plot.

FEATURE- Any labeled points are cleared.

◇ **Relocate a Group of Points** : Changes mouse cursor and enables selection of points by brushing.

FEATURE- The mouse cursor changes to a cross hair. User must then left-click drag a box over their desired points to move. Upon release of the left mouse button, the user is prompted to select the central point of the new location, i.e. the new location for the center of the brushed box. The next left click will select this point and relocate all selected points keeping their internal configuration intact. The mouse cursor is changed back to its original state. Changes are also reflected in the • Shepard Plot and stress value in the • Information Table.

◇ **Remove a Point** : Enables removal of single point by mouse cursor. Changes are also reflected in the • Shepard Plot and stress value in the • Information Table.

FEATURE- Upon selection, the mouse cursor will change to a cross hair and the user is prompted to select the point they would like to remove. Left clicking the button will remove the point and change the mouse cursor to its original state. All removed points are listed in the ◇ Removed Points table of the • Information Table.

◇ **Use Coordinates as Starting Configuration** : Alters the • MDS Configuration Plot. Only applicable when active configuration is result of ○ Metric Symmetric SMACOF, ○ Least Squares Scaling, ○

Non-Metric Symmetric SMACOF, ○ Kruskal's Analysis or ○ Sammon Mapping.

FEATURE– Will perform the version of MDS used to produce the configuration and use the current configuration as the starting configuration for the process.

◇ **Rotate and Reflect** : Displays the Rotate and Reflect window.

MENU– Provides means of performing rotation and reflection operations on the configuration.

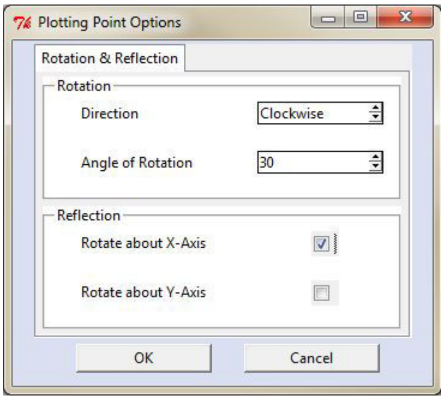


Figure 10: Rotate and Reflect

Options for Rotate and Reflect

Direction : Selection made by scrollbox.

FEATURE– Choice of rotating points either clockwise or anticlockwise

Angle of Rotation : Selection made by scrollbox.

FEATURE– Choice of degrees by which reflection must be. Zero for no rotation.

Reflect about X-Axis : Selection made with checkbox.

FEATURE– Will flip configuration top to bottom.

Reflect about Y-Axis : Selection made with checkbox.

FEATURE– Will flip configuration left to right.

OK : Selection Button.

FEATURE– Upon selection the • MDS Configuration Plot will have the above changes made.

Cancel : Selection Button.

FEATURE– Exit menu with no changes.

◇ **Advanced Zoom** : Displays the Advanced Zoom window.

MENU– Provides user with means of performing the zoom function on the • MDS Configuration Plot in a more advanced and specific fashion than by using the '+' and '-' buttons.

Options for Advanced Zoom

Zoom Ratio : Selection made by scrollbox.

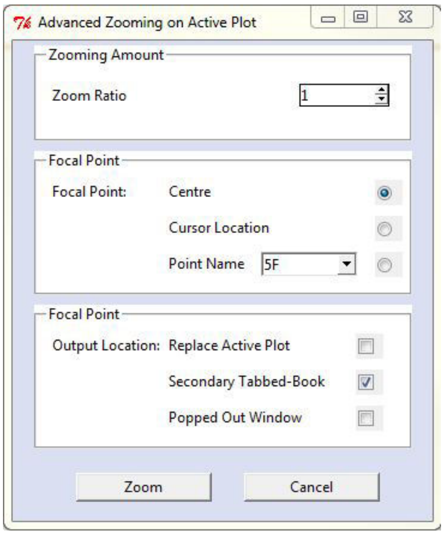


Figure 11: Advanced Zoom

FEATURE- Determines the amount by which the zoom is performed.

Focal Point: Center : Selection made by radiobutton.

FEATURE- Selection will perform the zoom around the exact center of the configuration.

Focal Point: Cursor Location : Selection made by radiobutton.

FEATURE- Selection will perform the zoom at the point of next left click by the user in the configuration.

Focal Point: Point Name : Selection made by radiobutton and dropdown menu of point names.

FEATURE- Selection will perform zoom with a specific point as the focal point. Point is selected manually.

Output Location: Replace Active Plot : Selection made by radiobutton.

FEATURE- Selection will produce the zoom in the Main Plotting Area.

Output Loaction: Secondary Tabbed Book : Selection made by radiobutton.

FEATURE- Selection will produce the zoom as a • Zoomed Plot in Secondary Plotting Area.

Popped Out Window : Selection made by radiobutton.

FEATURE- Selection will produce the zoom in a popped out window.

Zoom : Selection Button.

FEATURE- Selection will perform the zoom with specification defined above.

Cancel : Selection Button.

FEATURE- Selection will exit the menu with no alterations performed.

◇ **Change Point Colour** : Enables selection of points via brushing.

FEATURE- Upon Selection user is prompted to draw a box around the points they would like to alter the colour of. Releasing the left click will call the native operating system colour selection window whereby selection will change the colour of the selected points. This change supersedes the colours defined according to the object categories.

◇ **Default Point Colours** : Reverts to original state.

FEATURE- Will change all point object colours to the default colour specified in the • Plot Options Menu.

◇ **Display Variable Axes** : Alters the • MDS Configuration Plot.

FEATURE- In the case where a $\mathbf{Z}$:$n \times m$ matrix was uploaded via ∗ Load Dataset, this feature will be available. Selection will display the m variable axes through the origin of the configuration. Each axis is assigned its own colour. Menu item is also a checkbox and indicates activation with a checkmark. De-selection removes variable axes.

◇ **Remove Axes of Variable(s)** : Opens the Variable Axes Removal Options window.

MENU- The menu provides ability to determine which subset of variable axes are displayed.

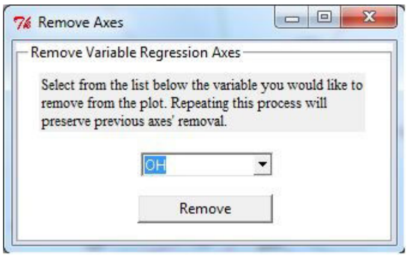


Figure 12: Variable Axes Removal Options

Options for Variable Axes Removal Options

Variable Selection : Selection by drop down menu.

FEATURE- The variable selected will be removed from the • MDS Configuration Plot. All removed variables are listed in the ◇ Removed Axes table of the • Information Table.

Remove : Selection Button.

FEATURE- Selection will perform the removal of the selected variable.

◇ **Pop-out Plot** : Opens an external plotting window.

FEATURE- The • MDS Configuration Plot will be popped out for external viewing in an external *tcltk* window. This second plot will display all visual aspects of the original version in the Main Plotting Area and reflect changes as they are made, but will lack the full extent of the capabilities itself.

◇ **Copy Plot to Clipboard** : Utilises native operating system clipboard capabilities.

FEATURE- The • MDS Configuration Plot figure is copied from the MDS-GUI to the computer's clipboard where it may be pasted and utilised externally.

◇ **Plot Options** : Opens the • Plot Options Menu menu.

• **Secondary Plot Menu** : Menu accessed via right clicking when the mouse is hovering over any of the plots housed in the Secondary Plotting Area area. All menus are similar with only slight differences. For demonstration purposes, only the menu corresponding to the • Shepard Plot will be described. Similar menus apply to the • Stress Plot, • Log Stress and • Scree Plot.

MENU- The secondary plot menu provide functions relating to the • Shepard Plot. The features utilised will alter the Shepard Plot itself and in some cases the • MDS Configuration Plot as well.

◇ **Label Specific Point** : Opens the Shepard Point Label.

MENU- The menu provides option to label a specific point from the Shepard Plot. Visually selecting a desired point is impossible since by default the Shepard Plot does not print point labels due to the inevitable large number of points.

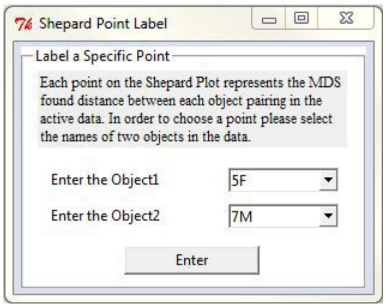


Figure 13: Shepard Point Label

Options for Shepard Point Label

Enter Object1 : Selection is via dropdown list of object names.

FEATURE- Selects the first object name index to be labeled.

Enter Object2 : Selection is via dropdown list of object names.

FEATURE- Selects the second object name index to be labeled.

Enter : Selection Button

FEATURE- Upon selection, the Shepard point relating to the object pairing defined above will be labled. This highlights the point on the • Shepard Plot and draws the corresponding line in the same colour between the defined objects on the • MDS Configuration Plot.

◇ **Clear Added Labels** : Alters the • Shepard Plot and the • MDS Configuration Plot.

FEATURE- Upon selection, all Shepard point labels that have been added will be cleared from the plot and all other plot features will be reverted to their original settings. In addition, corresponding object pairing lines will be removed from the configuration plot.

◇ **Pop-Out Enlarged Plot** : Opens an external plotting window.

FEATURE- The • Shepard Plot will be popped out for external viewing in an external *tcltk* window.

This second plot will display all visual aspects of the original version in the Secondary Plotting Area and reflect changes as they are made, but will lack the full extent of the capabilities itself.

◇ **Copy Shepard Plot to Clipboard** : Utilises native operating system clipboard capabilities.

FEATURE- The • Shepard Plot figure is copied from the MDS-GUI to the computer's clipboard where it may be pasted and utilised externally.

◇ **Shepard Plot Options** : Opens the relevant • Plot Options Menu.

• **General Settings** : The menu is recalled from the ∗ General Settings option of the ◇ General menu.

MENU-Controls various technical aspects of the MDS-GUI. The four tabs of the menu are the ◇ General Tab, ◇ Convergence Tab, ◇ Graphical Tab and ◇ Visualisation Tab.

◇ **General Tab** : First tab of • General Settings window.

MENU-The first tab of the menu contains *Computation Options* and *Windows Options*. By default, the MDS-GUI will calculate all five major processes automatically, these being: the • MDS Configuration Plot, the • Shepard Plot, the • Scree Plot, and (when applicable) the • Stress Plot and • Log Stress. In *Computation Options* the option is available to deactivate any of these computations for all subsequent use. This option is expected to be exercised when data is sizable and all computations have proven to be excessively time consuming.

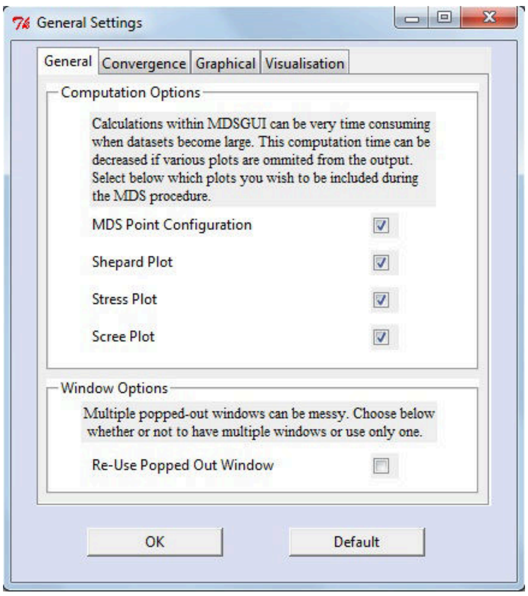


Figure 14: General Tab of General Settings

Options for General Tab of General Settings

Compute MDS Point Configuration : First option of Computation Options frame. Selection is made with checkbox.

FEATURE- When deselected, MDS processes will not compute or display the • MDS Configuration Plot. Default is on.

Compute Shepard Plot : Second option of Computation Options frame. Selection is made with checkbox.

FEATURE- When deselected, MDS processes will not compute or display the • Shepard Plot. Default is on.

Compute Stress Plots : Third option of Computation Options frame. Selection is made with checkbox.

FEATURE- When deselected, MDS processes will not compute or display the • Stress Plot or • Log Stress. Default is on.

Compute Scree Plot : Fourth option of Computation Options frame. Selection is made with checkbox.

FEATURE- When deselected, MDS processes will not compute or display the • Scree Plot. Default is on.

Re-use Popped Out Window First option of Windows Options frame. Selection is made with checkbox.

FEATURE- When selected, only one popped out plot can be visible at a time. Creating a new popped out plot will destroy the old and keep the new. When deselected, numerous popped out plots may exist. Default is off.

◇ **Convergence Tab** : Second tab of • General Settings window.

MENU-The *Convergence* tab contains options relating to the allocation of computational resources dedicated to the MDS procedures.

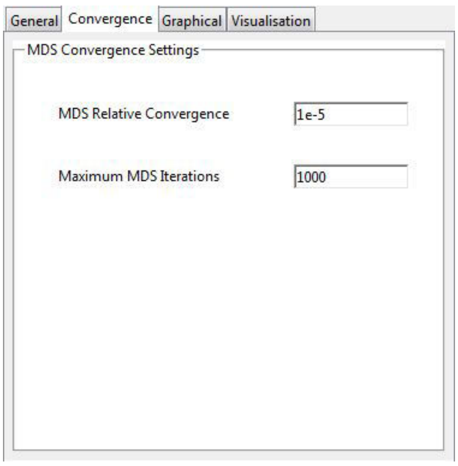


Figure 15: Convergence Tab of General Settings

Options for Convergence Tab of General Settings

MDS Relative Convergence : Numerical text input required.

`FEATURE-` When numeric figure is $\in [0:1]$, the MDS procedure will converge when the difference between stress values between iterations is less than or equal to the input amount.

Maximum MDS Iterations : Numerical text input required.

`FEATURE-`This input number must be greater than zero. Defines after how many iterations the MDS process should stop if convergence has not yet been met.

◇ **Graphical Tab** : Third tab of • General Settings window.

`MENU-`The *Graphical* tab, relates to default options of the visual MDS outputs. *Graphical Settings* is used in conjunction with the settings in • Plot Options Menu. *Point Label Settings* controls point labels manually added by the user to any relevant plot.

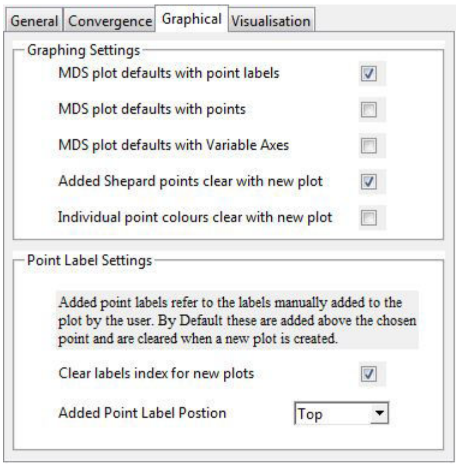


Figure 16: Graphical Tab of General Settings

Options for Graphical Tab of General Settings

MDS plot defaults with point labels : Selection is by checkbox.

`FEATURE-` New • MDS Configuration Plot will by default have labels.

MDS plot defaults with points : Selection is by checkbox.

`FEATURE-` New • MDS Configuration Plot will by default show points.

MDS plot defaults with Variable Axes : Selection is by checkbox.

`FEATURE-` New • MDS Configuration Plot will by default display all m variable axes.

Added Shepard Points clear with new plot : Selection is by checkbox.

`FEATURE-` New • MDS Configuration Plot and • Shepard Plot will by default show no prior labeled Shepard points.

Individual point colours clear with new plot : Selection is by checkbox.

`FEATURE-` New • MDS Configuration Plot will by default not carry through manual object colour changes (excluding category related colours).

Clear labels index for new plots : Selection is by checkbox.

FEATURE– New ● MDS Configuration Plot will by default not show previously labeled points.

Added Point Label Position : Selection is by dropdown menu.

FEATURE– All point labels on all ● MDS Configuration Plot and ● Shepard Plot will have their labels in this position relative to the point. Options are 'Top', 'Bottom', 'Left' or 'Right'. Default is 'Top'.

◇ **Visualisation Tab** : Fourth tab of ● General Settings window.

MENU–The *Visualisation* tab specifies the elements of the MDS procedures that should have its iterative nature depicted visually. This should not be confused with the *Computation Options* elements in the *General* tab. The outputs that have been unchecked will still be processed (provided they have not been deactivated), but will not have their respective plots updated after each iteration. In this case, the final result is plotted upon process completion.

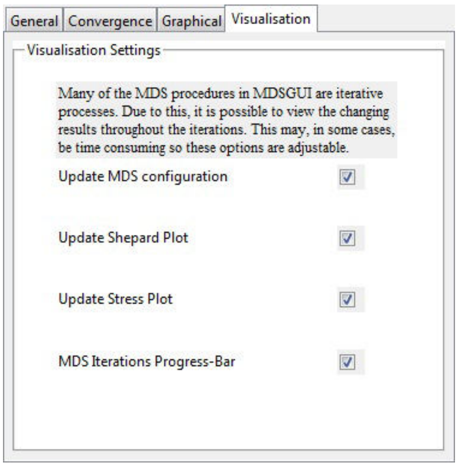


Figure 17: Visualisation Tab of General Settings

Options for Visualisation Tab of General Settings

Update MDS Configuration : Selection is by checkbox.

FEATURE– If selected, the ● MDS Configuration Plot is updated after every iteration of the active MDS process. Default is 'on'.

Update Shepard Plot : Selection is by checkbox.

FEATURE– If selected, the ● Shepard Plot is updated after every iteration of the active MDS process. Default is 'on'.

Update Stress Plots : Selection is by checkbox.

FEATURE– If selected, the ● Stress Plot and ● Log Stress are updated after every iteration of the active MDS process. Default is 'on'.

Updata iterations Progress-Bar : Selection is by checkbox.

`FEATURE-` If Selected, the Progress-Bar shown at the leftmost side of the Information Panel progresses throughout every iteration of the active MDS process and in the construction of the • Scree Plot. Default is 'on'.

• **Data Options Menu** : The menu is recalled from the ∗ Data Options option from the ⋄ Data menu.

`MENU-` The menu is similar to the New Active Dataset, with the difference that this menu may be called at any point and have the settings changed. The displayed tab is associated with an uploaded $\mathbf{Z}$:$n \times m$ matrix or $\Delta : n \times n$ matrix.

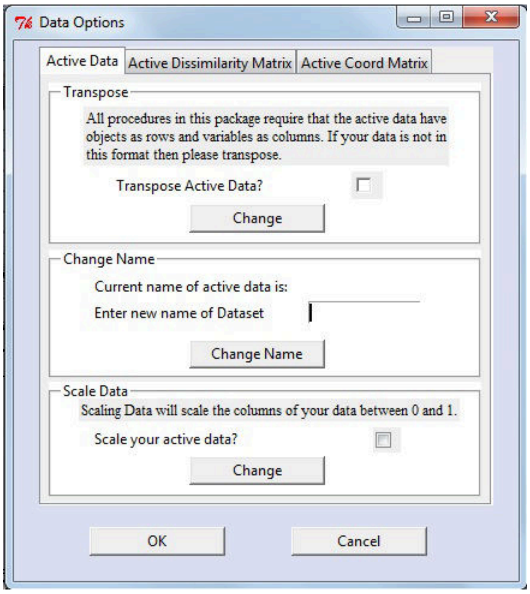


Figure 18: Data Options Menu

Options for Data Options Menu

Transpose Active Data : Selection by checkbox.

`FEATURE-` Applicable when uploaded data has variables as rows and objects as columns. Selection transposes data such that rows are objects and variables as columns.

Enter New Name of Dataset : Required Text Entry.

`FEATURE-` Name of data can be changed at any point. Result is reflected in the Information Panel and future exported documents and • MDS Configuration Plot.

Change Name : Selection Button.

`FEATURE-` Selection will only change name of data and nothing else.

Scale Your Active Data : Selection by checkbox.

FEATURE- Selection will scale data such that all columns range from zero to one.

Cahnge : Selection button.

FEATURE- Makes all specified changes.

• **MDS Options Menu** : The menu is recalled from the ∗ MDS Options option of the ◇ Multivariate Tools menu.

MENU- The purpose of menu is to provide key adjustments to the MDS procedure that effect the output in a substantial way. The three tabs of the menu are the ◇ Dimensions Tab, the ◇ Starting Configuration Tab and the ◇ Stress Tab.

◇ **Dimensions Tab** : The first tab of the • MDS Options Menu.

MENU-The tab adjusts the user defined p for all subsequent MDS procedures. When a new data set is added, the selection is populated with the entries being $1,2,...,n-1$.

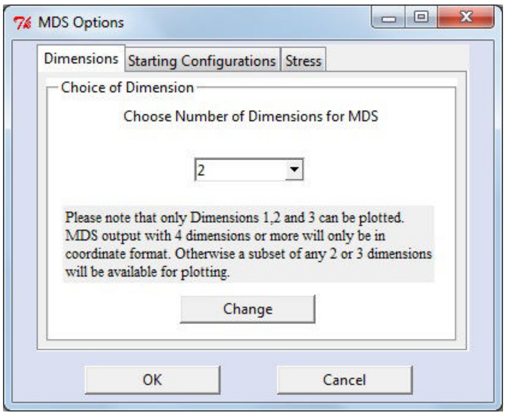


Figure 19: Dimensions Tab

Options for Dimensions Tab

Choice of dimension : Selection by drop down list of numbers.

FEATURE- All subsequent procedures will be performed in p dimensions. If $p = 1$ is chosen, a warning box will be displayed urging the user against it. Default is $p = 2$.

Change : Selection Button.

FEATURE- Activates the chosen p value. The result is indicated in the Information Panel.

◇ **Starting Configuration Tab** : The second tab of the • MDS Options Menu.

MENU-The *Starting Configurations* tab provides the user the option to change what starting configuration is used in the MDS procedures (where starting configuration is relevant). Three options are provided, being: the $n \times p$ result of Classical Scaling on data; a random configuration, where the $n \times p$ matrix is uniformly distributed and doubly centered around the origin; and finally a configuration in any of the five main plotting tabs may be set as the starting configuration for all subsequent procedures. Therefore, the user may use, say, the result of a Sammon Mapping procedure as the starting configuration for a SMACOF procedure.

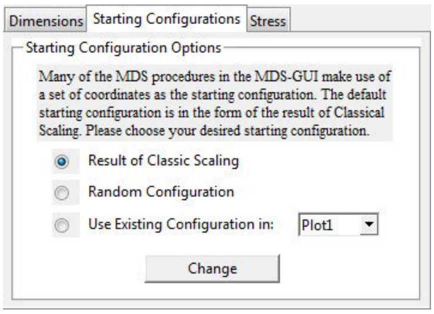


Figure 20: Starting Configurations Tab

Options for Starting Configurations Tab

Result of Classical Scaling : Selection is by radiobutton.

`FEATURE-` All MDS procedures requiring a starting configuration will now use the result of `cmdscale` with Δ as input. This is the default setting.

Random Configuration : Selection is by radiobutton.

`FEATURE-` All MDS procedures requiring a starting configuration will use an $\mathbf{X}$:$n \times p$ that is randomised using the uniform distribution.

Use Existing Configuration in : Selection is by radiobutton and drop down menu of Plotting Tabs options.

`FEATURE-` All MDS procedures requiring a starting configuration will use the $\mathbf{X}$:$n \times p$ from the • MDS Configuration Plot from the selected plotting area.

Change : Selection Button.

`FEATURE-` Selection will change the starting configuration as specified.

◇ **Stress Tab** : The third tab of the • MDS Options Menu.

`MENU-`The final tab of *MDS Options* is called *Stress* and controls the measure of stress used to assess the goodness-of-fit of all configurations. It should be noted that this does not affect the loss function used within each MDS functions, as each method is usually defined by their specific loss function. The stress method chosen here simply defines how the accuracy of the final configurations are measured, in order to compare accuracy of configurations in absolute terms, as they need to compared on an identical scale.

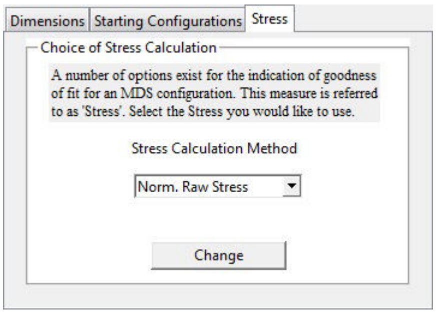


Figure 21: Stress Tab

Options for Stress Tab

Stress Calculation Method : Selection by drop down menu.

`FEATURE-` Result will adjust the stress value shown in the ◇ MDS Configurations table of the • Information Table, and the stress values in the • Stress Plot, • Log Stress and • Scree Plot. Options for this include: Normalised Raw Stress (Default), STRESS1, STRESS2 and Pearson's Correlation Coefficient.

Change : Selection Button.

`FEATURE-`Selection will change the stress calculation method as specified.

- **Plot Options Menu** : The menu is recalled from the ◇ Plot Options option of the • Main Plot Menu.

  `MENU-` Each plotting area found in the MDS-GUI has a Plot Options menu, which may be accessed via a right click of the plot and selecting the *Plot Options* option. These areas include: • MDS Configuration Plot, • Shepard Plot, • Stress Plot, • Log Stress, • Scree Plot, • Zoomed Plot, • RGL 3D Plot, • Static 3D Plot and • Procrustes Analysis Plot. Each of these menu's are set out in a similar way with only slight differences depending on the nature of the plot itself. The menu associated with the • MDS Configuration Plot will be used for demonstrative purposes. Four tabs exist on the majority of the menus.

  ◇ **General Tab** : First tab of the • Plot Options Menu.

  `MENU-` The *General* tab of the plot menu deals with overall settings of the plotting area.

Options for Plot Options: General Tab

Display Main Title : Selection by checkbox.

`FEATURE-` When active, the • MDS Configuration Plot displays a title at the top center, indicating the type of MDS performed and the name of the data. Default is 'on'.

Display Distance Measure : Selection by checkbox.

`FEATURE-` When active, the • MDS Configuration Plot displays (at the top left) the current distance metric used to calculate Δ, as selected in ∗ Dissimilarity Matrix Calculation.

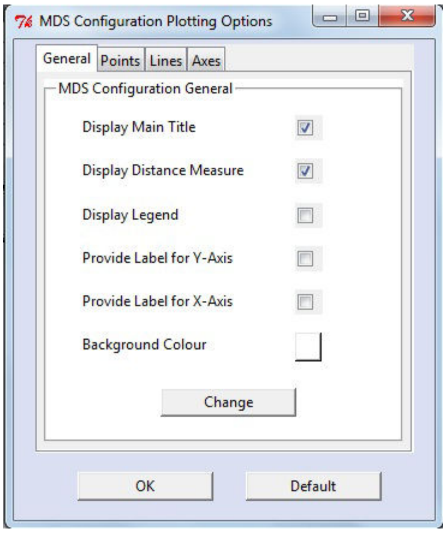


Figure 22: Plot Options: General Tab

Default is 'on'.

Provide Label for Y-Axis : Selection by checkbox.

FEATURE- When selected, a small dialog box appears prompting user to name the axis. The result is added vertically along the Y-Axis. Default is 'off'.

Provide Label for X-Axis : Selection by checkbox.

FEATURE- When selected, a small dialog box appears prompting user to name the axis. The result is added horizontally along the X-Axis. Default is 'off'.

Background Colour : Selection by clicking coloured box, calling native operating system colour selection window.

FEATURE- Selection will cause the entire backdrop of the • MDS Configuration Plot to change to the desired colour.

Change : Selection Button.

FEATURE- Makes all specified changes.

◇ **Points Tab** : Second tab of the • Plot Options Menu.

MENU- The visual effects of the n points making up the configuration are controlled by the *Points* tab.

Options for Plot Options: Points Tab

Display Points : Selection by checkbox.

FEATURE- When selected each of the n objects in the • MDS Configuration Plot will display a specified symbol at the corresponding exact coordinates of the **X** matrix. Default is 'off'.

Display Point Labels : Selection by checkbox.

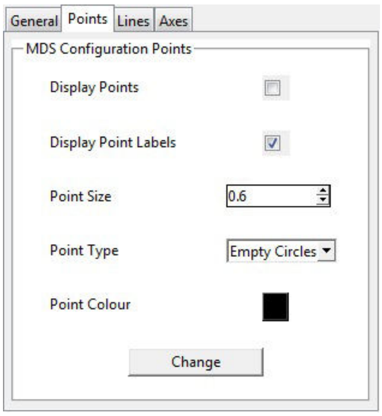


Figure 23: Plot Options: Points Tab

FEATURE- When selected each of the n objects in the • MDS Configuration Plot will display the object name. If points are also activated, the label will be at the position specified in the ◇ Graphical Tab of • General Settings. If points are not activated, the label is displayed at the exact coordinates. Default is 'on'.

Point Size : Selection by scrollbox.

FEATURE- Adjusts the 'cex' par function. Default is '0.6'.

Point Type : Selection by dropdown menu.

FEATURE- Adjusts the 'pch' par function for points (if selected). Options include, 'Solid Dot', 'Empty Dot', 'Solid Block', 'Empty Block', 'Solid Triangle', 'Empty Triangle' and 'Cross'.

Point Colour : Selection by clicking coloured box, calling native operating system colour selection window.

FEATURE- All objects will be changed to the specified colour.

Change : Selection Button.

FEATURE- Makes all specified changes.

◇ **Lines Tab** : Third tab of the • Plot Options Menu.

MENU- The lines tab controls the various forms of line additions that may be added to the configuration plot.

Options for Plot Options: Lines Tab

Display Regression Axes : Selection by checkbox.

FEATURE- Refers to the variable axes regression lines. Toggles their display on and off. Seen as an alternative to the ◇ Display Variable Axes option of the • Main Plot Menu. Default is 'off'.

Display Distance Lines Between Points Corresponding to Shepard Points : Selection by

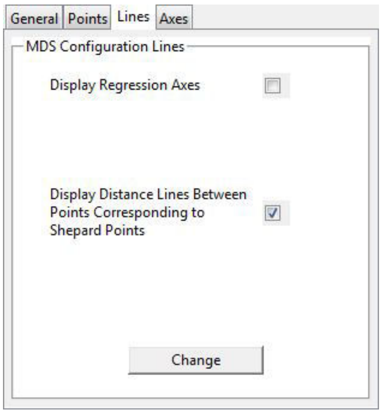


Figure 24: Plot Options: Lines Tab

checkbox.

FEATURE– Toggles on and off the display of lines between objects that correspond to the labeled points on the • Shepard Plot. Default is 'on'.

Change : Selection Button.

FEATURE– Makes all specified changes.

◇ **Axes Tab** : Fourth tab of the • Plot Options Menu.

MENU– The measurement indicators on the axes of the plotting area for the configuration are usually regarded as irrelevant. This is due to the fact that only the relative distances between points is useful which may be observed visually. This being said, the option to add the numerical axes measures is available to the user if that output is desirable to them.

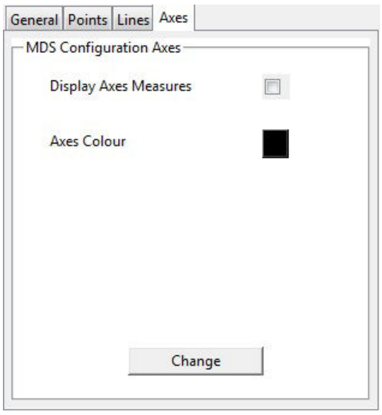


Figure 25: Plot Options: Axes Tab

Options for Plot Options: Axes Tab

Display Axes Measures : Selection by checkbox.

FEATURE– When selected, the numerical points along both axes are displayed on the •

MDS Configuration Plot. Default is ‘off’.

Axes Colour : Selection by clicking coloured box, calling native operating system colour selection window.

`FEATURE-` Selected colour will be reflected on the axes border of the plot as well as the measures if they have been added.

Change : Selection Button.

`FEATURE-` Makes all specified changes.

Other Features

- **NotesScript** : The Notes and scripting capabilities of the MDS-GUI are housed in the Notes/Script tab of the Plotting Tabs.

  `MENU-` The tab in the secondary plotting area named *Notes/Script* serves two purposes within the MDS-GUI. The first is a note making service available to the user where it acts as a simple text input location. The area comes complete with copy and pasting functionality, either through a right click menu or the keyboard shortcuts Ctrl-C and Ctrl-V. The second feature is the interface that has been customised between the tab and the active *R*-Environment. This link allows the user to treat the area as a scripting location, whereby they may write their *R* code and run it directly from the MDS-GUI to the console.

  ◇ **Run as Code** : Selection Button

  `FEATURE-` Treats all text in the text box as code and runs it through the *R* console.

  ◇ **Save Notes** : Selection Button

  `FEATURE-` Opens the native operating system save file window. Allows user to save contents to a txt file for external use.

  ◇ **Load Notes** : Selection Button

  `FEATURE-` Opens the native operating system load file window. Allows user to load contents of an external txt file.

- **Information Table** : the table section of the MDS-GUI summarises all relevant information of each of the five plotting areas of Plotting Tabs. This allows the user to make direct comparisons from a numerical point of view.

  ◇ **MDS Configurations** : Every time an MDS procedure is performed, all relevant information is updated on the table in the row corresponding to the active plotting area.

  `FEATURE-` Informs of the MDS method of each plot.

  `FEATURE-` Informs of the distance metric of each plot.

  `FEATURE-` Informs of the Stress value of each plot.

  `FEATURE-` Informs of the MDS method of each plot.

  `FEATURE-` Informs of the tolerance used for the process shown in each plot.

  `FEATURE-` Informs of the number of iterations used in the MDS process for each plot.

◇ **Removed Points** : When objects have been removed from any of the Plotting Tabs, the object names are displayed in this table.

FEATURE- The table corresponds to the active plotting tab. Each tab has its own individual table.

FEATURE- Each element of the table will be coloured the same as they were in the • MDS Configuration Plot.

FEATURE- Each element of the table may be returned to the • MDS Configuration Plot by right clicking the cell and selecting 'Replace Point in Active Cell'.

◇ **Removed Axes** : When variable axes have been removed from any of the Plotting Tabs, the variable names are displayed in this table.

FEATURE- The table corresponds to the active plotting tab. Each tab has its own individual table.

FEATURE- Each element of the table will be coloured the same as they were in the • MDS Configuration Plot.

FEATURE- Each element of the table may be returned to the • MDS Configuration Plot by right clicking the cell and selecting 'Replace Point in Active Cell'.

• **Animated Optimisation** : Refers to the ability to see real time changes in the • MDS Configuration Plot, • Shepard Plot, • Stress Plot and • Log Stress during MDS procedures.

FEATURE- By replotting all plots after every iteration, the illusion is of fluid motion of the points and lines (if computer has sufficient capabilities). The feature works best when datasets are smallest.

FEATURE- The feature may be turned off for every plot individually from the ◇ Visualisation Tab of • General Settings.

FEATURE- Only the focused tab in the Secondary Plotting Area will be updated to avoid unnecessary utilisation of computational power.

FEATURE- All popped out plots will also update.

◇ **End Process** : Selection button found at the left most side of the Information Panel.

FEATURE- If a process is taking too long, as may be the case when the data is large, the button may be selected to stop the MDS process and return GUI functionality to the user.

• **Keyboard Shortcuts** : The following shortcuts have been programmed to the keyboard.

◇ **1:** Focuses Plot1 of the Plotting Tabs

◇ **2:** Focuses Plot2 of the Plotting Tabs

◇ **3:** Focuses Plot3 of the Plotting Tabs

◇ **4:** Focuses Plot4 of the Plotting Tabs

◇ **5:** Focuses Plot5 of the Plotting Tabs

◇ **←:** Adjusts the displayed axes of the active • MDS Configuration Plot to the left (configuration moves right).

◇ **→:** Adjusts the displayed axes of the active • MDS Configuration Plot to the right (configuration moves left).

◇ ↑: Adjusts the displayed axes of the active • MDS Configuration Plot upwards (configuration moves down).

◇ ↓: Adjusts the displayed axes of the active • MDS Configuration Plot downwards (configuration moves up).

◇ +: Zooms in the active • MDS Configuration Plot around the center of the displayed axes.

◇ −: Zooms out the active • MDS Configuration Plot around the center of the displayed axes.

◇ **c:** Returns the axes of the active • MDS Configuration Plot to its original orientation.

◇ **w:** Applicable only to the • Static 3D Plot. Rotates plotting area around its horizontal axes leftwards.

◇ **s:** Applicable only to the • Static 3D Plot. Rotates plotting area around its horizontal axes rightwards.

◇ **a:** Applicable only to the • Static 3D Plot. Extends horizontal axes outwards.

◇ **d:** Applicable only to the • Static 3D Plot. Reduces horizontal axes inwards.

◇ **C:** Applicable only to the • Static 3D Plot. Returns plot to its original orientation.

◇ **Ctrl-C:** Applicable only to • NotesScript. Copies all highlighted text to clipboard.

◇ **Ctrl-V:** Applicable only to • NotesScript. Pastes text from clipboard to text area.

◇ **Ctrl-L:** Performs the ∗ Load Dataset operations.

Known Issues

- Versions of $R > 2.13.0$ up to current version (2.15.1) have trouble displaying all text in some of the popped out menu windows. Specifically, the Load Dataset menu and the 3D Plotting window. This is a suspected consequence of a certain error with the interaction of **tcltk2** and the R platform. The problem is expected to rectify itself with future updates.

Remainder of list is still being compiled.

B.3 MDS-GUI Vignette

Some R packages are accompanied by at Vignette, which serves to provide further information regarding the use of the software. These documents often are focused more on interpretation of results and include example results from included data. The following document is the Vignette that accompanies the MDS-GUI along with the User Manual. This Vignette first provides brief information on Multidimensional Scaling and then demonstrates the use of the MDS-GUI with the use of the Morse-Code data and the the Breakfast Cereal Data. More information is provided in the Introduction of the document.

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

August 11, 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 de 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 six 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. The MDS-GUI however handles all necessary input management automatically, and the user may simply choose their desired method regardless of how the data was uploaded.

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, 2012b). 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, 2012a), 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.

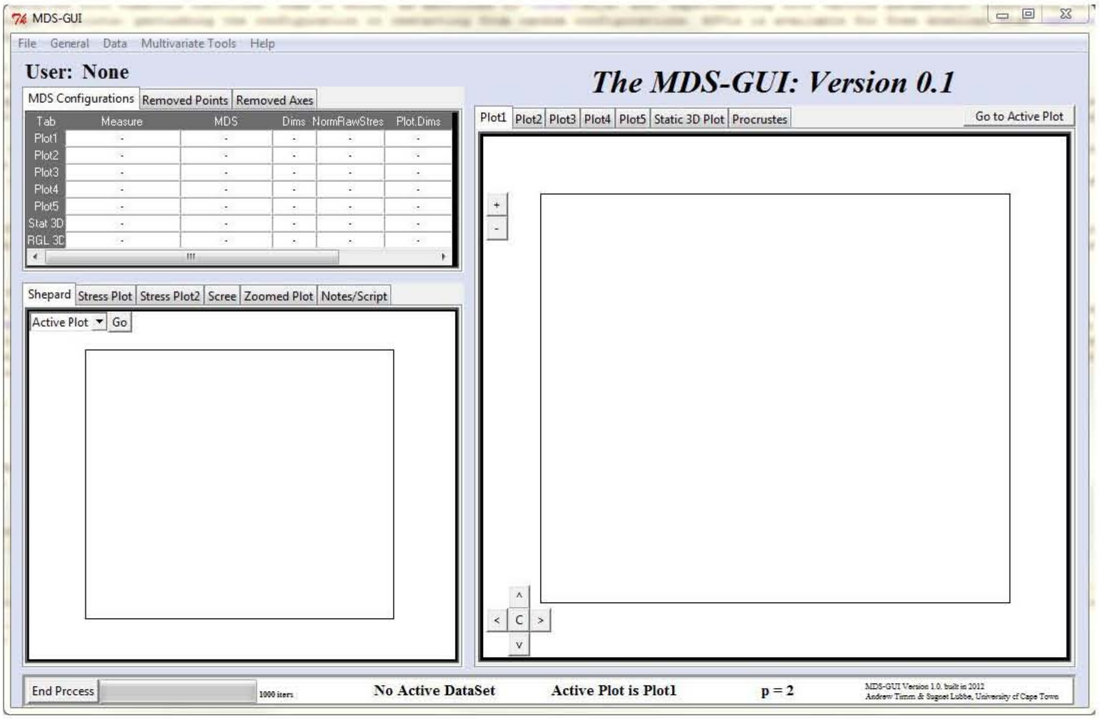


Figure 1: The MDS-GUI

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 Similarity 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.

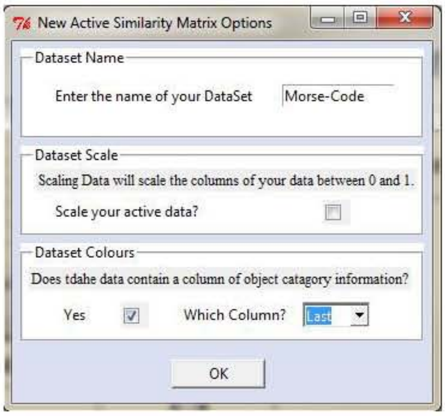


Figure 2: New Active Dataset Options: Similarity Matrix

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

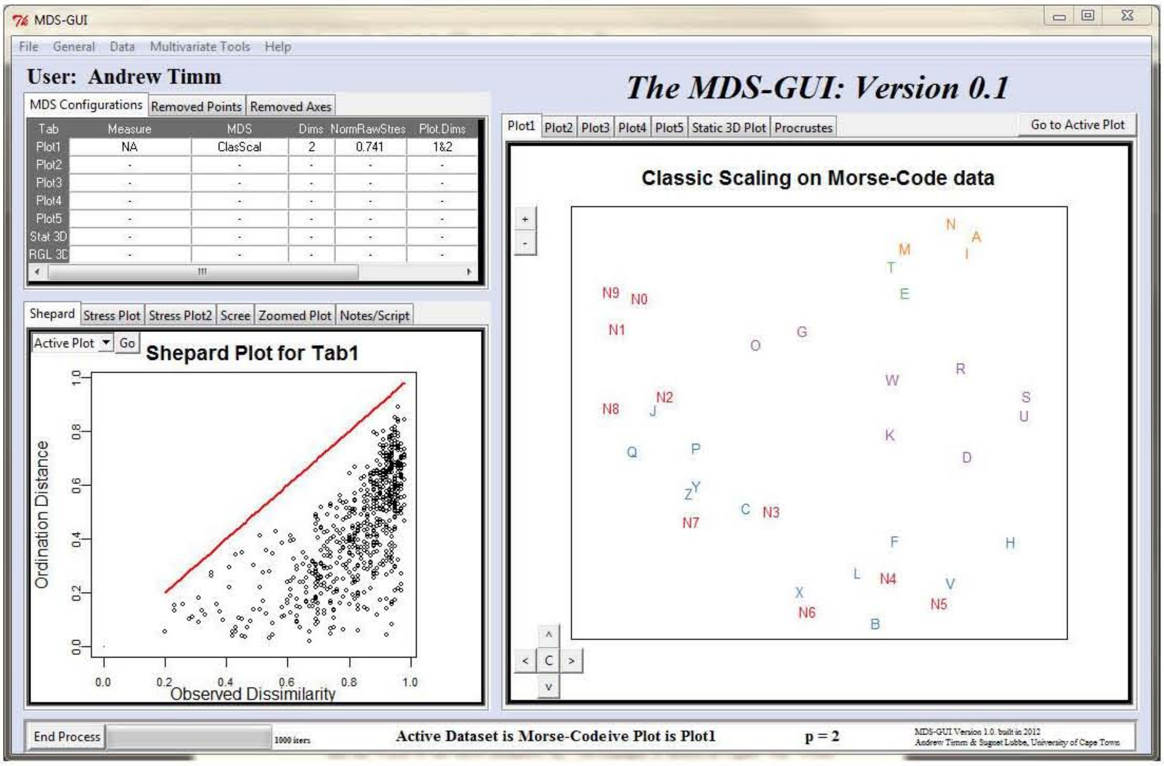


Figure 3: Morse-Code: Classical Scaling

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* $\rightarrow$ *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* $\rightarrow$ *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.

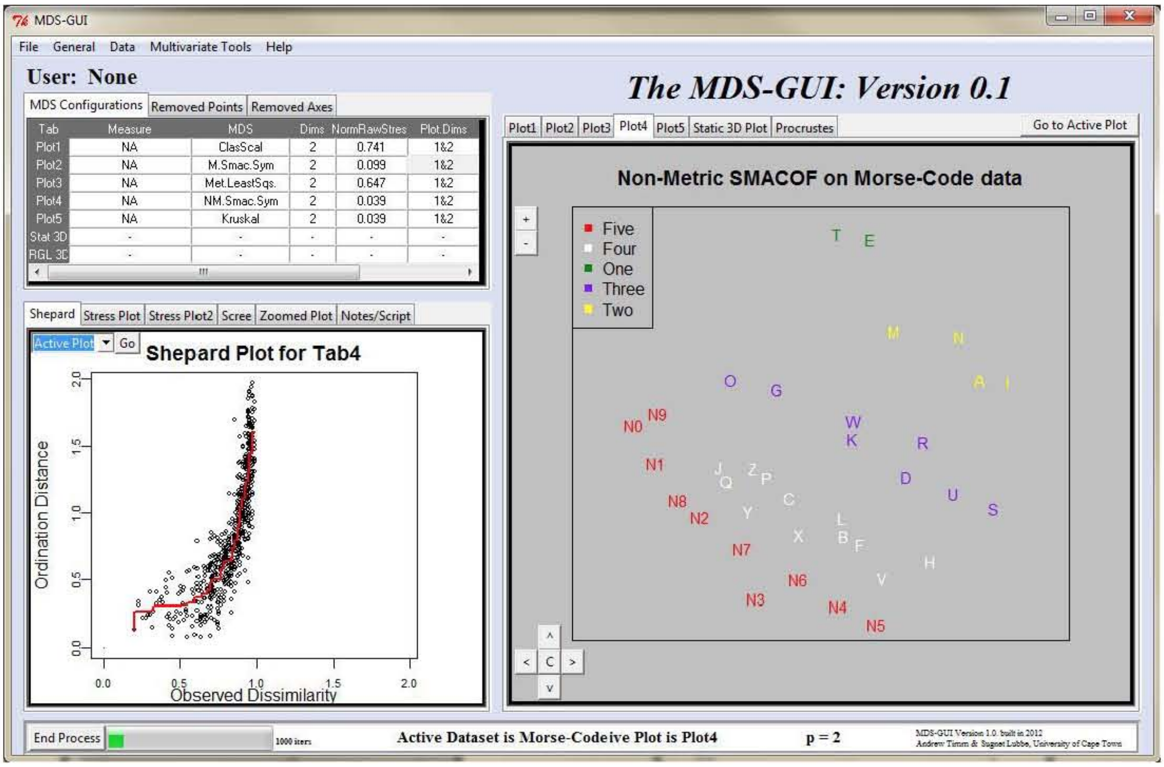


Figure 4: Morse-Code: Non-Metric SMACOF

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 Breakfast Cereal Data: A step by step beginners guide

1. **Upload the Cereal Data**: The breakfast cereal data is in the form of an $n \times m$ $\mathbf{Z}$ 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.

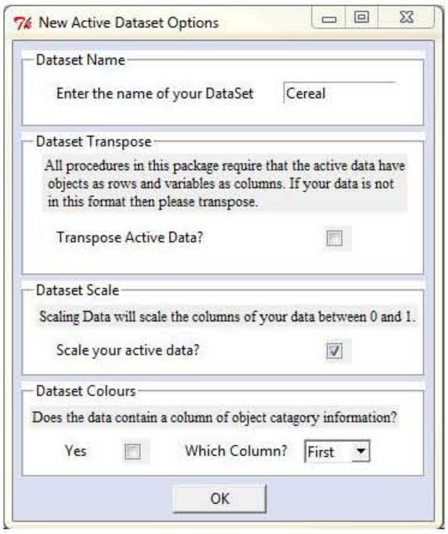


Figure 5: New Active Dataset Options

2. **Repeat Analysis as before**

3. **This time...**

(a) **Experiment with Different Metric Calculations:** Now that an $n \times m$ $\mathbf{Z}$ 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

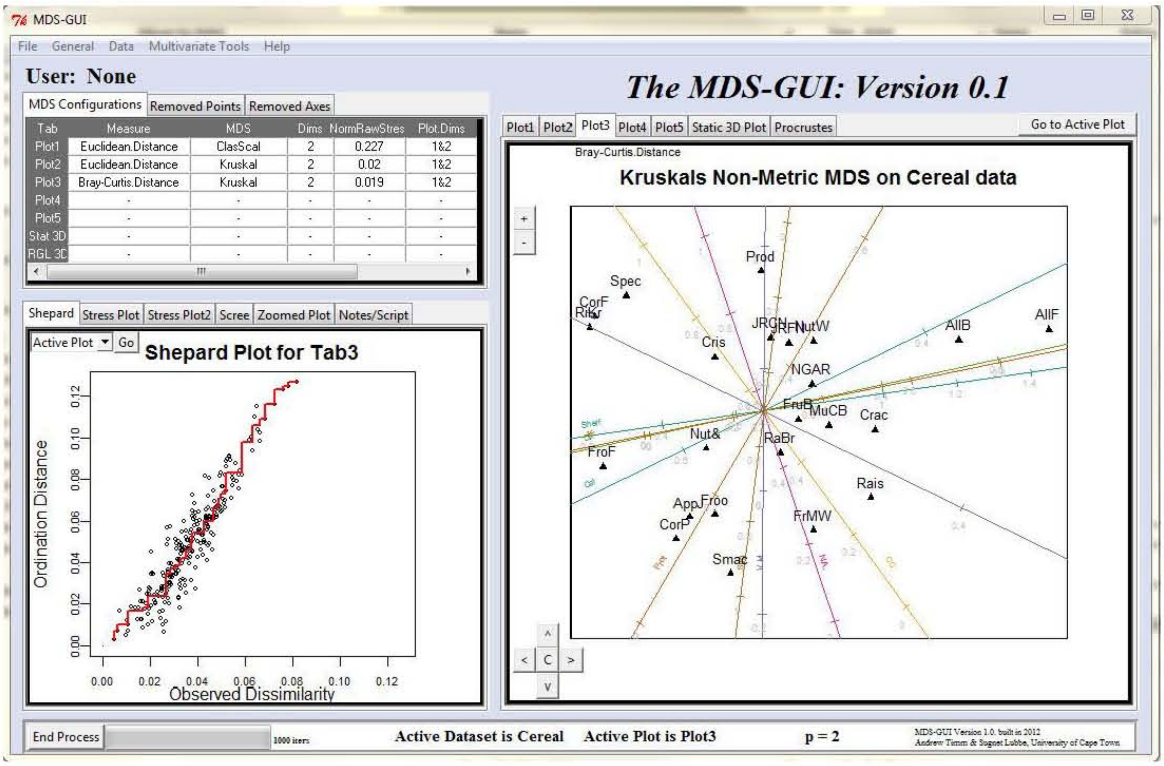


Figure 6: Cereal: Kruskal's Analysis with Bray Curtis Metric

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* $\rightarrow$ *MDS Options* $\rightarrow$ *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

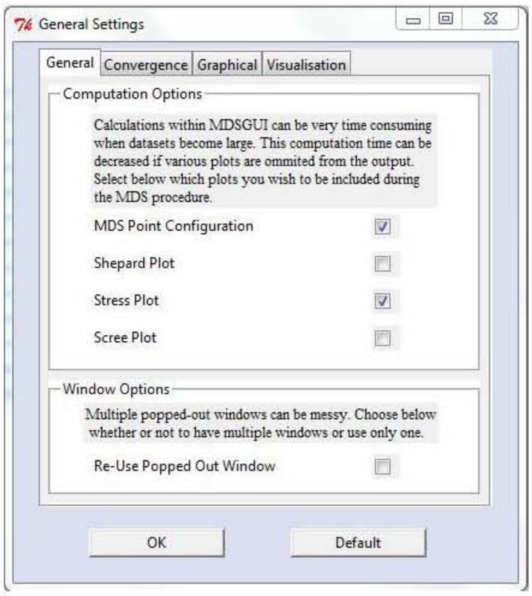


Figure 7: General Settings Menu

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* $\rightarrow$ *General Settings* $\rightarrow$ *General Tab*. De-select Shepard Plot and Scree Plot as shown in Figure 7. Click ‘OK’.