

University of Strathclyde

Department of Mathematics and  
Statistics

Multivariate Analysis of  
Metabonomic Data

Alexios Prelondjos

Degree of Doctor of Philosophy

2014

This thesis is the result of the author' s original research. It has been composed by the author and has not been previously submitted for examination which has led to the award of a degree.

©: The copyright of this thesis belongs to the author under the terms of the United Kingdom Copyright Acts as qualified by University of Strathclyde Regulation 3.50. Due acknowledgement must always be made of the use of any material contained in, or derived from, this thesis.

Signed:

Date:

# Abstract

Metabonomics is one of the main technologies used in biomedical sciences to improve understanding of how various biological processes of living organisms work. It is considered a more advanced technology than e.g. genomics and proteomics, as it can provide important evidence of molecular biomarkers for the diagnosis of diseases and the evaluation of beneficial adverse drug effects, by studying the metabolic profiles of living organisms. This is achievable by studying samples of various types such as tissues and biofluids. The findings of a metabonomics study for a specific disease, disorder or drug effect, could be applied to other diseases, disorders or drugs, making metabonomics an important tool for biomedical research.

This thesis aims to review and study various multivariate statistical techniques which can be used in the exploratory analysis of metabonomics data. To motivate this research, a metabonomics data set containing the metabolic profiles of a group of patients with epilepsy was used. More specifically, the metabolic fingerprints (proton NMR spectra) of 125 patients with epilepsy, of blood serum type, have been obtained from the Western Infirmary, Glasgow, for the purposes of this project. These data were originally collected as baseline data in a study to investigate if the treatment with Anti-Epileptic Drugs (AEDs), of patients with pharmaco-resistant epilepsy affects the seizure levels of the patients. The response to the drug treatment in terms of the reduction in seizure levels of these patients enabled two main categories of response to be identified, i.e. responders and the non-responders to AEDs. We explore the use of statistical methods used in metabonomics to analyse these data. Novel aspects of the thesis are the use of Self Organising Maps (SOM) and of Fuzzy Clustering Methods to pattern recognition in metabonomics data.

Part I of the thesis defines metabonomics and the other main "omics" technologies, and gives a detailed description of the metabonomics data to be analysed, as well as a description of the two main analytical chemical techniques, *Mass Spectrometry* (MS) and *Nuclear Magnetic Resonance Spectroscopy* (NMR), that can be used to generate metabonomics data. Pre-processing and pre-treatment methods that are commonly used in NMR-generated metabonomics data to enhance

the quality and accuracy of the data, are also discussed.

In Part II, several unsupervised statistical techniques are reviewed and applied to the epilepsy data to investigate the capability of these techniques to discriminate the patients according to their type of response. The techniques reviewed include *Principal Components Analysis* (PCA), *Multi-dimensional scaling* (both *Classical scaling* and *Sammon's non-linear mapping*) and *Clustering* techniques. The latter include *Hierarchical clustering* (with emphasis on *Agglomerative Nesting* algorithms), *Partitioning methods* (*Fuzzy* and *Hard* clustering algorithms) and *Competitive Learning* algorithms (*Self Organizing maps*). The advantages and disadvantages of the different methods are examined, for this kind of data. Results of the exploratory multivariate analyses showed that no natural clusters of patients existed with regards to their response to AEDs, therefore none of these techniques was capable of discriminating these patients according to their clinical characteristics.

To examine the capability of an unsupervised technique such as PCA, to identify groups in such data as the data based on metabolic fingerprints of patients with epilepsy, a simulation algorithm was developed to run a series of experiments, covered in Part III of the thesis. The aim of the simulation study is to investigate the extent of the difference in the clusters of the data, and under what conditions this difference is detectable by unsupervised techniques. Furthermore, the study examines whether the existence or lack of variation in the mean-shifted variables affects the discriminating ability of the unsupervised techniques (in this case PCA) or not.

In each simulation experiment, a reference and a test data set were generated based on the original epilepsy data, and the discriminating capability of PCA was assessed. A test set was generated by mean-shifting a pre-selected number of variables in a reference set. Three methods of selecting the variables to mean-shift (*maximum* and *minimum standard deviations* and *maximum means*), five subsets of variables of sizes 1, 3, 20, 120 and 244 (total number of variables in the data sets) and three sample sizes (100, 500 and 1000) were used. Average values in 100 runs of an experiment for two statistics, i.e. the *misclassification rate* and the *average separation* (Webb, 2002) were recorded. Results showed that the number of mean-shifted variables (in general) and the methods used to select the variables (in some cases) are important factors for the discriminating ability of PCA, whereas the sample size of the two data sets does not play any role in the experiments (although experiments in large sample sizes showed greater stability in the results for the two statistics in 100 runs of any experiment). The results have implications for the use of PCA with metabonomics data generally.



# Acknowledgements

As I am writing down the final, but very important part of my thesis, that of fondly remembering all those people that accompanied and supported me during this long journey, it becomes in my mind even more important considering that in the last few years my country, Greece, has suffered and is still suffering from one of the worst crises in its modern history. This crisis, is not only economic, but also political, democratic, and more importantly a crisis of human ethics, dignity and ideals. Thus, having to deal with the everyday ramifications of the situation back home, the following people played a more important than usual role in helping me to successfully finishing my thesis.

Firstly, I would like to say TAPADH LEIBH to the whole Department of Mathematics and Statistics, for their continuous support, patience and understanding all these years, and also for making me feel comfortable from the first moment of my arrival.

I would also like to thank my supervisors Dr. Alison Gray and Professor Chris Robertson, for their time and effort to provide me with all the important help and support, so that this project was made possible, and also for their patience and understanding.

I am very grateful to the late Professor George Gettinby for his useful advice and support during the project, as well as his patience and understanding.

Many thanks go to Dr. John Parkinson for the generation of the NMR data used in the research of this project, and its pre-processing and general preparation.

I would also like to thank Ian Thurlbeck for all the invaluable computer help.

Many thanks to Dr. David Greenhalgh and Dr. David Morris, for their patience and for putting up with a noisy Greek like me, all these years.

Also, I would like to thank Dr. David Morris, Dr. Andrew Wade and Dr. Steven Corson for the random chat during the project. Dr. David Morris also for his important advice on specific aspects in the creation and better readability of objects such as figures. Dr. Stephen Corson also for his invaluable input and help in solving some problems with the appearance of my references and the

bibliography, as well as for his help with the printing of my thesis.

Special thanks go to my friend and colleague Dr. Abdullah Almarashi, for the random chat and letting me rant, as well as for all his help with my transportation needs in the shopping and other activities, and for the nice meals. In addition, for helping me to find accommodation in the last year of my project.

Last but not least, I am grateful to my parents, Lewis and Helen, for their continuous love, patience and support, and together with Georgia and my best friends in Greece Stathis and Maria Georgoulis, for keeping me company through the very long phone or skype calls during all these years, helping me to keep my sanity during the long hours of work.

Especially to my father (Emeritus Professor of Physics, Technological Educational Institution of Athens - TEI Athens), I devote this work, for his invaluable help in making some of the most important decisions in my life, and for being the greatest factor in what I am today ...

# Contents

<b>I</b>	<b>Project and Data Description</b>	<b>1</b>
	<b>Introduction</b>	<b>2</b>
<b>1</b>	<b>Generating Information About the Genome</b>	<b>4</b>
1.1	Introduction . . . . .	4
1.2	Scope of the Thesis . . . . .	5
1.3	Bionomics . . . . .	5
1.3.1	Genomics/Transcriptomics . . . . .	5
1.3.2	Proteomics . . . . .	6
1.3.3	Metabonomics . . . . .	6
1.3.4	Advantages of Metabonomics . . . . .	7
1.3.5	Metabolic Profiling, Fingerprinting and Target Analysis . . . . .	8
1.3.6	Metabonomics Applications . . . . .	8
1.4	Toxicogenomics . . . . .	9
1.5	Chemometrics . . . . .	9
<b>2</b>	<b>Project Description</b>	<b>11</b>
2.1	Introduction . . . . .	11
2.2	Definition of Epilepsy . . . . .	11
2.2.1	Types of Epileptic Seizures . . . . .	12
2.2.2	Epilepsies and Epileptic Syndromes . . . . .	13
2.2.3	Epidemiology of Epilepsy . . . . .	14
2.3	Description of the Problem . . . . .	14
2.4	Data Description . . . . .	15
2.4.1	Clinical Information of the Patients . . . . .	15
2.4.2	The Data Set . . . . .	16
2.4.3	Characteristics of Subjects in the Current Data Set . . . . .	18
<b>3</b>	<b>Metabonomics - Analytical Techniques</b>	<b>20</b>
3.1	Introduction . . . . .	20
3.2	Mass Spectrometry (MS) . . . . .	21
3.2.1	Definition . . . . .	21
3.2.2	Theoretical Background of MS . . . . .	22
3.2.3	History of MS . . . . .	23
3.2.4	Description of Mass Spectrometers . . . . .	24
3.2.4.1	Ionisation Process . . . . .	27
3.2.4.2	Mass Analysis . . . . .	29

---

3.2.4.3	Detection of the Ions . . . . .	32
3.2.5	Mass Spectrum . . . . .	32
3.2.6	Applications of MS to Metabonomics Studies . . . . .	33
3.3	Nuclear Magnetic Resonance Spectroscopy (NMR Spectroscopy) . . . . .	34
3.3.1	Introduction . . . . .	34
3.3.2	Theoretical Background of NMR . . . . .	34
3.3.3	History of NMR . . . . .	36
3.3.4	Description of a NMR Spectrometer . . . . .	37
3.3.4.1	The Magnet . . . . .	38
3.3.4.2	The Detector . . . . .	39
3.3.5	Description of an NMR Spectrum . . . . .	39
3.3.6	Applications of NMR to Metabonomics Studies . . . . .	40
3.4	Comparison of Metabonomics Techniques . . . . .	41
<b>4</b>	<b>Pre-processing and Pre-treatment of the Data</b> . . . . .	<b>43</b>
4.1	Introduction . . . . .	43
4.2	Sensitivity of a Measurement in NMR . . . . .	44
4.3	Pre-processing Methods . . . . .	44
4.3.1	Binning (Bucketing) . . . . .	44
4.3.2	Baseline Correction . . . . .	46
4.3.3	Deconvolution . . . . .	46
4.3.4	Smoothing . . . . .	48
4.4	Pre-treatment Methods . . . . .	48
4.4.1	Introduction . . . . .	48
4.4.2	Scaling . . . . .	48
4.4.3	Transformations . . . . .	52
	<b>Summary</b> . . . . .	<b>59</b>
	<b>II Pattern Recognition - Unsupervised Techniques</b> . . . . .	<b>62</b>
	<b>Introduction</b> . . . . .	<b>63</b>
<b>5</b>	<b>Principal Components Analysis</b> . . . . .	<b>65</b>
5.1	Introduction . . . . .	65
5.2	Theoretical Background . . . . .	66
5.2.1	Testing Data Suitability for PCA . . . . .	68
5.2.2	Determining the Number of Components to Extract . . . . .	69
5.3	Application of PCA in the Epilepsy Data . . . . .	70
5.3.1	Introduction . . . . .	70
5.3.2	Data Suitability . . . . .	72
5.3.3	Identification of the Number of Components to Retain . . . . .	72
5.3.4	Results of PCA . . . . .	75
5.4	Conclusions . . . . .	90

<b>6</b>	<b>Multidimensional Scaling</b>	<b>93</b>
6.1	Introduction	93
6.2	Classical Scaling	94
6.3	Metric MDS	96
6.3.1	Introduction	96
6.3.2	Metric Least - Squares (LS) Scaling	96
6.3.2.1	Sammon's Non-linear Mapping (NLM)	96
6.4	Application of MDS to the Epilepsy Data	97
6.4.1	Introduction	97
6.4.2	Classical Scaling Solution	97
6.4.3	Sammon's Non-linear Mapping (NLM) Solution	104
6.5	Conclusions	109
<b>7</b>	<b>Cluster Analysis</b>	<b>112</b>
7.1	Introduction	112
7.2	Clustering Considerations and Decisions	113
7.3	Proximity Measures	114
7.4	The Silhouette Coefficient	116
7.5	Hierarchical Clustering Methods	117
7.5.1	Introduction	117
7.5.2	Agglomerative Nesting Algorithms	117
7.5.3	Divisive Clustering Algorithms	122
7.5.4	Application of HCA to the Epilepsy Data	123
7.5.4.1	Introduction	123
7.5.4.2	Comparison of Hierarchical Clustering Results	124
7.5.4.3	Identification of the Optimal Number of Clusters	131
7.5.4.4	Identification of the Best Method for the <i>Response to AEDs</i> Information	133
7.5.4.5	Investigation on Other Clinical Characteristics	142
7.6	Partitioning Methods	150
7.6.1	Introduction	150
7.6.2	Fuzzy Clustering Algorithms	150
7.6.2.1	Introduction	150
7.6.2.2	The Fanny Fuzzy Clustering Algorithm	152
7.6.2.3	Fuzziness of a Clustering Solution	152
7.6.3	Application of Fuzzy Clustering to the Epilepsy Data	153
7.6.3.1	Introduction	153
7.6.3.2	Comparison of Fuzzy Clustering Methods	154
7.6.3.3	Identification of the Optimal Number of Clusters	155
7.6.3.4	Discrimination of the Clinical Characteristics	158
7.6.4	Hard Clustering Algorithms	165
7.6.4.1	Introduction	165
7.6.4.2	The $k$ -means Clustering Algorithm	166
7.6.4.3	Clustering Criteria	168
7.6.5	Application of the $k$ -means Algorithm to the Data	170
7.6.5.1	Introduction	170
7.6.5.2	Determination of the Optimal Number of Clusters	170

7.6.5.3	Discrimination of the Clinical Characteristics . . .	172
7.7	Competitive Learning Algorithms . . . . .	175
7.7.1	Introduction . . . . .	175
7.7.2	Self Organizing Maps . . . . .	175
7.7.2.1	Introduction . . . . .	175
7.7.2.2	Classic On-line SOM Algorithm . . . . .	176
7.7.2.3	Classic Batch SOM Algorithm . . . . .	178
7.7.2.4	Goodness of Mapping . . . . .	179
7.7.2.5	Means of Visualization . . . . .	180
7.7.2.6	Application to the Epilepsy Data . . . . .	183
7.8	Conclusions . . . . .	195
	<b>Summary</b>	<b>199</b>
	<b>III Simulation Experiments</b>	<b>202</b>
<b>8</b>	<b>Data Simulation</b>	<b>203</b>
8.1	Introduction . . . . .	203
8.2	Supervised Learning Techniques . . . . .	205
8.2.1	Introduction . . . . .	205
8.2.2	Supervised Classification . . . . .	205
8.2.3	Two Class Classifiers . . . . .	206
8.2.4	Bayes Decision Rule . . . . .	206
8.2.5	Discriminant Functions . . . . .	208
8.2.5.1	Linear Discriminant Functions . . . . .	209
8.3	Simulation Procedure . . . . .	210
8.3.1	Preparation of the Epilepsy Data . . . . .	210
8.3.2	Generation of the Reference Data Set . . . . .	210
8.3.3	Generation of a Test Data Set . . . . .	212
8.3.4	Row-scaling of Data Sets . . . . .	214
8.3.5	Column-scaling of Data Sets . . . . .	215
8.3.6	Principal Component Analysis . . . . .	217
8.3.6.1	PCA Scores and <i>Average Separation Plots</i> . . . . .	218
8.3.7	Linear Discriminant Analysis (LDA) . . . . .	220
8.3.8	Simulation Algorithm . . . . .	220
8.4	Simulation Experiments and Results . . . . .	222
8.4.1	Introduction . . . . .	222
8.4.2	Case MS244 . . . . .	222
8.4.3	Maximum Deviation (MAXDEV) . . . . .	225
8.4.4	Minimum Deviation (MINDEV) . . . . .	247
8.4.5	Maximum Mean (MAXMEAN) . . . . .	249
8.5	Conclusions . . . . .	250
<b>9</b>	<b>Conclusions and Further Work</b>	<b>258</b>
9.1	Conclusions . . . . .	258
9.2	Further Recommendations . . . . .	267

---

<b>Appendices</b>	<b>271</b>
<b>Appendix A - Lists of Mean-Shifted Variables</b>	<b>272</b>
<b>Appendix B - Simulation Results</b>	<b>276</b>
Appendix B.1 - MINDEV Results . . . . .	277
Appendix B.2 - MAXMEAN Results . . . . .	291
<b>Appendix C - R Code Used in the Simulation Algorithm</b>	<b>305</b>
Appendix C.1 - List of R Functions Used in the Project . . . . .	306
Appendix C.2 - Simulation Algorithm . . . . .	308
<b>Appendix D - Vignette for the Simulation Algorithm</b>	<b>339</b>
D.1 Data Input and Pre-treatment . . . . .	340
D.2 Generation of Reference and Test Sets . . . . .	343
D.3 Simulation Analyses . . . . .	344
D.4 Execution of the Simulation Algorithm . . . . .	347
D.5 Illustrate the Effect of Mean-shifting . . . . .	350
D.6 Plot PCs Scores and LDA Boundary . . . . .	353
D.7 Plot Statistics vs Offsets . . . . .	354
D.8 Plot Additional Information . . . . .	356
<b>Appendix E - Lists of Components of Mass Spectrometers</b>	<b>358</b>
<b>Bibliography</b>	<b>360</b>

# List of Tables

<b>1</b>	<b>Generating Information About the Genome</b>	<b>4</b>
<b>2</b>	<b>Project Description</b>	<b>11</b>
2.1	Seizure type of patients in the current data set before and after simplification . . . . .	17
2.2	Clinical characteristics of the patients in the reduced data set . . .	18
<b>3</b>	<b>Metabonomics - Analytical Techniques</b>	<b>20</b>
3.1	Examples of nuclei and their spin . . . . .	35
<b>4</b>	<b>Pre-processing and Pre-treatment of the Data</b>	<b>43</b>
<b>5</b>	<b>Principal Components Analysis</b>	<b>65</b>
5.1	Variance information for the first ten PCs . . . . .	72
5.2	Comparison of various stopping rules . . . . .	75
5.3	Coefficients for the PCR model of <i>Seizure Type</i> . . . . .	83
5.4	Coefficients for the PCR model of <i>Response to AEDs</i> . . . . .	85
<b>6</b>	<b>Multidimensional Scaling</b>	<b>93</b>
6.1	$P_k$ and <i>Mardia</i> criteria for various Minkowski metrics in <i>classical scaling</i> - $k = 2$ . . . . .	98
<b>7</b>	<b>Cluster Analysis</b>	<b>112</b>
7.1	Interpretation of the silhouette coefficient values . . . . .	117
7.2	Clustering techniques from the general formula of Jambu (1978) . . .	122
7.3	Agglomerative coefficients for the epilepsy data . . . . .	125
7.4	<i>Cophenetic correlation</i> for the 28 hierarchical clustering methods . .	128
7.5	<i>Gower distance</i> for the 28 hierarchical clustering methods . . . . .	131
7.6	Cross-tabulation of the 2-cluster partitions for <i>Response to AEDs</i> . .	133
7.7	Cross-tabulation of 2-cluster partition to <i>Response to AEDs</i> . . . .	135
7.8	Cross-tabulation of 3-6 cluster partitions to <i>Response to AEDs</i> . . .	139
7.9	$\chi^2$ test for homogeneity of the <i>Response to AEDs</i> . . . . .	142
7.10	Cross-tabulation of 2-6 cluster partitions to <i>Gender</i> and <i>Age</i> . . . .	144
7.11	$\chi^2$ test for homogeneity of the <i>Gender</i> . . . . .	144
7.12	Cross-tabulation of 2-6 clusters partitions with <i>Seizure Type</i> and <i>BMI</i> . . . . .	147
7.13	$\chi^2$ test for homogeneity of the <i>Seizure Type</i> . . . . .	147
7.14	Kruskal-Wallis Rank Sum test of the <i>Age</i> and <i>BMI</i> . . . . .	149
7.15	Comparison of fuzzy clustering methods with regards to pre-selected <i>fuzzifier</i> values and distance measures . . . . .	154



7.16	Comparison of fuzzy clustering methods with regards to pre-selected number of clusters and distance measures . . . . .	156
7.17	Membership coefficients for the fuzzy 2-cluster partition . . . . .	157
7.18	Cross-tabulation of the optimal 2-cluster fuzzy partition to <i>Response to AEDs</i> . . . . .	160
7.19	Cross-tabulation of the optimal 2-cluster fuzzy partition to <i>Gender and Age</i> . . . . .	160
7.20	Cross-tabulation of the optimal 2-cluster fuzzy partition to <i>Seizure Type and BMI</i> . . . . .	162
7.21	Cross-tabulation of the optimal 2-cluster <i>k</i> -means partition to <i>Response to AEDs</i> . . . . .	174
7.22	Number of patients in each of the six clusters returned by SOM and the clinical characteristics <i>Gender, Seizure type and Response to AEDs</i> . . . . .	191
7.23	Number of patients in each of the six clusters returned by SOM and the clinical characteristics <i>Age and Body-Mass-Index (BMI)</i> . . . . .	191
7.24	$\chi^2$ and Kruskal-Wallis tests for the clinical characteristics . . . . .	194
<b>8</b>	<b>Data Simulation</b>	<b>203</b>
8.1	List of simulation experiments . . . . .	214
8.2	Average stats for case MS244 . . . . .	224
8.3	Coefficient of variation results for case MS244 . . . . .	227
8.4	Summary results (offsets) for the <i>LDA misclassification rates</i> in the case MS244 . . . . .	228
8.5	Average stats for case MS120 with method MAXDEV . . . . .	229
8.6	Coefficient of variation results - case MS120 using method MAXDEV . . . . .	232
8.7	Average stats for case MS20 with method MAXDEV . . . . .	235
8.8	Coefficient of variation results for case MS20 using method MAXDEV . . . . .	236
8.9	Average stats for case MS3 with method MAXDEV . . . . .	238
8.10	Coefficient of variation results for case MS3 using method MAXDEV . . . . .	241
8.11	Average stats for case MS1 with method MAXDEV . . . . .	243
8.12	Coefficient of variation results for case MS1 using method MAXDEV . . . . .	245
8.13	Summary results (offsets) for the <i>LDA misclassification rates</i> in all MS cases for MAXDEV . . . . .	246
8.14	Summary results for offsets . . . . .	255
8.15	Summary results for the <i>average separation</i> . . . . .	256
8.16	Summary results for offsets - Row-Scaled Data . . . . .	257
<b>9</b>	<b>Conclusions and Further Work</b>	<b>258</b>
9.1	Comparison of scaling techniques . . . . .	259
9.2	Proportions of the total variance explained by the first three principal components for the three data sets . . . . .	259
9.3	Summary results (offsets) for the various MS cases with S500 and using MAXMEAN (apart from MS244) with the UNSCALED and LOG-TRANSFORMED data . . . . .	265

9.4	Summary results (offsets) for the various MS cases with S500 and using MAXMEAN (apart from MS244) with the ROW-SCALED and LOG-TRANSFORMED data . . . . .	266
9.5	Interpretation of the derived PCs in the two PCA methods . . . . .	267
<b>Appendix A - Lists of Mean-Shifted Variables</b>		<b>272</b>
A.1	Mean-Shifted Variables for the MAXDEV cases . . . . .	273
A.2	Mean-Shifted Variables for the MINDEV cases . . . . .	274
A.3	Mean-Shifted Variables for the MAXMEAN cases . . . . .	275
<b>Appendix B - Simulation Results</b>		<b>276</b>
B.1	Average stats for case MS120 with method MINDEV . . . . .	278
B.2	Average stats for case MS20 with method MINDEV . . . . .	279
B.3	Average stats for case MS3 with method MINDEV . . . . .	280
B.4	Average stats for case MS1 with method MINDEV . . . . .	281
B.5	Coefficient of variation results for case MS120 using method MINDEV	287
B.6	Coefficient of variation results for case MS20 using method MIN- DEV . . . . .	288
B.7	Coefficient of variation results for case MS3 using method MIN- DEV . . . . .	289
B.8	Coefficient of variation results for case MS1 using method MIN- DEV . . . . .	290
B.9	Average stats for case MS120 with method MAXMEAN . . . . .	292
B.10	Average stats for case MS20 with method MAXMEAN . . . . .	293
B.11	Average stats for case MS3 with method MAXMEAN . . . . .	294
B.12	Average stats for case MS1 with method MAXMEAN . . . . .	295
B.13	Coefficient of variation results for case MS120 using method MAX- MEAN . . . . .	301
B.14	Coefficient of variation results for case MS20 using method MAX- MEAN . . . . .	302
B.15	Coefficient of variation results for case MS3 using method MAX- MEAN . . . . .	303
B.16	Coefficient of variation results for case MS1 using method MAX- MEAN . . . . .	304
<b>Appendix C - R Code Used in the Simulation Algorithm</b>		<b>305</b>
C.1	List of R functions written for the project . . . . .	307
<b>Appendix E - Lists of Components of Mass Spectrometers</b>		<b>358</b>
E.1	Comparison of <i>ionisation sources</i> . . . . .	359
E.2	Comparison of <i>mass analysers</i> . . . . .	359
E.3	Comparison of the most commonly used <i>detectors</i> . . . . .	359

# List of Figures

<b>1</b>	<b>Generating Information About the Genome</b>	<b>4</b>
1.1	Relationship between -omics . . . . .	6
<b>2</b>	<b>Project Description</b>	<b>11</b>
2.1	Clinical data for five of the subjects - cases 5,36,59,66 and 114 . .	16
2.2	An example of a dataset for bins from 10.98 - 10.70 . . . . .	17
2.3	Graphical representation of the most important characteristics of patients in the current data set . . . . .	18
<b>3</b>	<b>Metabonomics - Analytical Techniques</b>	<b>20</b>
3.1	Important timelines and contributions to Mass Spectrometry . . .	24
3.2	Schematic of the main components of a mass spectrometer . . . .	25
3.3	<i>Liquid chromatography</i> (LC) separation procedure . . . . .	26
3.4	Diagram of a <i>gas chromatography</i> mass spectrometer (GC-MS) . .	26
3.5	Diagram of a <i>capillary electrophoresis</i> mass spectrometer (CE-MS)	27
3.6	Basic diagram of an <i>electron</i> ionisation (EI) source . . . . .	27
3.7	Basic diagram of an <i>electrospray</i> ionisation (ESI) source . . . . .	28
3.8	Basic diagram of a <i>matrix assisted laser desorption</i> ionisation source	29
3.9	Single quadrupole mass analyser . . . . .	30
3.10	Triple quadrupole mass analyser . . . . .	30
3.11	Iontrap mass analyser . . . . .	31
3.12	Time-of-Flight mass analyser . . . . .	31
3.13	Example of a Mass Spectrum . . . . .	32
3.14	Spin energy levels . . . . .	36
3.15	The main components of a NMR spectrometer . . . . .	37
3.16	Main parts of a NMR magnet . . . . .	38
3.17	Example of a proton NMR spectrum . . . . .	40
<b>4</b>	<b>Pre-processing and Pre-treatment of the Data</b>	<b>43</b>
4.1	Example of binning the epilepsy data . . . . .	45
4.2	Illustration of baseline correction of the epilepsy metabolite spectra	47
4.3	PC1 vs PC2 scores plots for the scaled epilepsy data sets. . . . .	53
4.4	PC1 vs PC2 loadings plots for the scaled epilepsy data sets. . . .	54
4.5	PC1 vs PC2 scores plots for the transformed epilepsy data sets. .	56
4.6	PC1 vs PC2 loadings plots for the transformed epilepsy data sets.	57
<b>5</b>	<b>Principal Components Analysis</b>	<b>65</b>
5.1	Percentages of the total variation in the data explained by the first ten components. . . . .	73

5.2	Stopping rules for the number of components. . . . .	74
5.3	Scores plots for the epilepsy data for the first four PCs . . . . .	77
5.4	Outlier diagnostic plots using the score ( <i>SD</i> ) and the orthogonal distance ( <i>OD</i> ). . . . .	79
5.5	Percentages of the total variation in the data explained by the first ten components (after removing the outliers). . . . .	80
5.6	Scores plots for the epilepsy data superimposed with the <i>Gender</i> information . . . . .	81
5.7	Scores plots for the epilepsy data superimposed with the <i>Seizure type</i> information . . . . .	82
5.8	Scores plots for the epilepsy data superimposed with the <i>Response to AEDs</i> information . . . . .	84
5.9	Scores plots for the epilepsy data superimposed with the <i>Age</i> information . . . . .	86
5.10	Scores plots for the epilepsy data superimposed with the <i>BMI</i> information . . . . .	87
5.11	Loadings plots of the first four PCs for the epilepsy data . . . . .	89
<b>6</b>	<b>Multidimensional Scaling</b> . . . . .	<b>93</b>
6.1	Two-dimensional solution of <i>classical MDS</i> . . . . .	98
6.2	Minimum spanning tree for the two MDS configurations . . . . .	99
6.3	Two-dimensional solution of <i>classical MDS</i> superimposed with the <i>Gender</i> information . . . . .	100
6.4	Two-dimensional solution of <i>classical MDS</i> superimposed with the <i>Seizure Type</i> information . . . . .	101
6.5	Two-dimensional solution of <i>classical MDS</i> superimposed with the <i>Response to AEDs</i> information . . . . .	101
6.6	Two-dimensional solution of <i>classical MDS</i> superimposed with the <i>Age</i> information . . . . .	102
6.7	Two-dimensional solution of <i>classical MDS</i> superimposed with the <i>BMI</i> information . . . . .	103
6.8	Two-dimensional solution of <i>NLM MDS</i> . . . . .	104
6.9	Quality assessment of the two NLM solutions . . . . .	105
6.10	Two-dimensional solution of NLM superimposed with the <i>Gender</i> information . . . . .	106
6.11	Two-dimensional solution of NLM superimposed with the <i>Seizure Type</i> information . . . . .	107
6.12	Two-dimensional solution of NLM superimposed with the <i>Response to AEDs</i> information . . . . .	107
6.13	Two-dimensional solution of NLM superimposed with the <i>Age</i> information . . . . .	108
6.14	Two-dimensional solution of NLM superimposed with the <i>BMI</i> information . . . . .	109
<b>7</b>	<b>Cluster Analysis</b> . . . . .	<b>112</b>
7.1	Single and Complete linkage strategies . . . . .	118

7.2	Banner plot for the 2-clustering partition derived by Ward method using the Maximum distance metric . . . . .	126
7.3	<i>Shepard</i> -like diagrams comparing six <i>Cophenetic</i> to original distances	127
7.4	<i>Dendrogram</i> for the 2-cluster partition derived by the <i>Euclidean - Single linkage</i> clustering method . . . . .	129
7.5	<i>Dendrogram</i> for the 2-cluster partition derived by the <i>Maximum - Centroid</i> clustering method . . . . .	130
7.6	<i>Average silhouette widths</i> for partitions of 2-96 clusters for the two selected clustering methods . . . . .	132
7.7	Graphs of the <i>fusion level</i> values of the corresponding dendrograms to the two clustering methods . . . . .	134
7.8	<i>Silhouette plot</i> for the 2-cluster partition derived by the <i>Maximum - Average</i> clustering method . . . . .	136
7.9	<i>Silhouette plot</i> for the 2-cluster partition derived by the <i>Maximum - Ward</i> clustering method . . . . .	137
7.10	Scores plots of the first two PCs, superimposed with the 2-cluster partition derived by the <i>Maximum - Ward</i> clustering method . . .	138
7.11	<i>Dendrogram</i> for the 2-cluster partition derived by the <i>Maximum - Ward</i> clustering method . . . . .	140
7.12	<i>Heat map</i> of the distance matrix of the <i>Maximum - Ward</i> clustering method according to the dendrogram of Figure 7.11 . . . . .	141
7.13	Scores plots of the first two PCs, superimposed with the cluster partitions for 3-6 clusters, derived by the <i>Maximum - Ward</i> clustering method . . . . .	143
7.14	Scores plots of the first two PCs, superimposed with the cluster partitions for 3-6 clusters, derived by the <i>Maximum - Ward</i> clustering method and the <i>Age</i> information . . . . .	146
7.15	Scores plots of the first two PCs, superimposed with the cluster partitions for 3-6 clusters, derived by the <i>Maximum - Ward</i> clustering method and the <i>BMI</i> information . . . . .	148
7.16	<i>Average silhouette widths</i> for partitions of 2-47 clusters for the selected fuzzy clustering method ( <i>SqEuclidean</i> metric and <i>fuzzifier</i> value 2) . . . . .	158
7.17	<i>Silhouette plot</i> for the 2-cluster partition derived by the <i>SqEuclidean - fuzzifier 2</i> fuzzy clustering method . . . . .	159
7.18	Scores plots of the first two PCs, superimposed with the 2-cluster partition derived by the <i>SqEuclidean - Fuzzifier 2</i> fuzzy clustering method . . . . .	161
7.19	Scores plots of the first two PCs, superimposed with the 2-cluster <b>fanny</b> partition, derived by the <i>SqEuclidean - Fuzzifier 2</i> clustering method and the <i>Age</i> information . . . . .	163
7.20	Scores plots of the first two PCs, superimposed with the 2-cluster <b>fanny</b> partition, derived by the <i>SqEuclidean - Fuzzifier 2</i> clustering method and the <i>BMI</i> information . . . . .	164
7.21	Values of four clustering criteria for <i>k</i> -means partitions of 2-10 clusters. . . . .	171

7.22	<i>Average silhouette widths</i> for partitions of 2-96 clusters for the selected <i>k</i> -means clustering method . . . . .	172
7.23	<i>Silhouette plot</i> for the 2-cluster partition derived by the <i>k</i> -means clustering method . . . . .	173
7.24	Visualization of SOM data projection - 2D rectangular grid. . . . .	176
7.25	Example of a Unified Distance matrix. . . . .	181
7.26	Example of a hit histogram. . . . .	182
7.27	Examples of component planes . . . . .	182
7.28	Convergence of the neighbourhood width function for the selected map . . . . .	184
7.29	Illustration of the quality of mapping with regards to the samples . . . . .	185
7.30	Unified Distance matrix for the $24 \times 2$ grid. . . . .	185
7.31	Hit histogram for the $24 \times 2$ SOM solution. . . . .	186
7.32	Illustration of clustering the epilepsy data to six groups using SOM . . . . .	186
7.33	Mean spectra for the six groups of patients. . . . .	188
7.34	Illustration of the mapping of the samples according to the patients' clinical characteristics . . . . .	189
7.35	Component planes for selected variables in the blood serum of the epilepsy patients, labelled by the chemical shift . . . . .	190
7.36	Scores plots of the first two PCs, superimposed with the 6-cluster partition derived by the SOM clustering model and the information of the five clinical characteristics . . . . .	193
<b>8</b>	<b>Data Simulation</b> . . . . .	<b>203</b>
8.1	Illustration of an LDA decision boundary for the original epilepsy data . . . . .	207
8.2	Geometry of a linear discriminant function given by equation 8.2.5 . . . . .	209
8.3	Illustration of the effect of log-transforming the epilepsy data . . . . .	211
8.4	Comparison of the original epilepsy mean spectrum (brown) to the generated mean spectrum (blue) using the positive definite covariance matrix obtained from the epilepsy data . . . . .	213
8.5	Illustration of the mean spectra of the UNSCALED and the ROW-SCALED data sets for the case MS244 . . . . .	216
8.6	The original epilepsy spectra before (top) and after (bottom) row-scaling . . . . .	217
8.7	The row and column (mean-centred) scaled epilepsy spectra . . . . .	218
8.8	Example of a statistics versus offsets plot. . . . .	219
8.9	Illustration of the mean-shifting procedure in the case MS244 with S500 and offset 20.09 . . . . .	223
8.10	Visualisation of the LDA boundaries in the case MS244 . . . . .	225
8.11	Graphical representation of the relation among <i>LDA misclassification rates</i> , <i>average separation</i> and offsets in the case MS244 . . . . .	226
8.12	Illustration of the mean-shifting procedure with method MAXDEV in the case MS120 with S500 and offset 1.55 . . . . .	228
8.13	Visualisation of the LDA boundaries for the two artificial data sets in the case MS120 (MAXDEV) . . . . .	230

8.14	Graphical representation of the relation among <i>LDA misclassification rates, average separation</i> and offset in the case MS120 for method MAXDEV . . . . .	231
8.15	Illustration of the mean-shifting procedure with method MAXDEV in the case MS20 with S500 and offset 2.18 . . . . .	233
8.16	Visualisation of the LDA boundaries for the two artificial data sets in the case MS20 (MAXDEV) . . . . .	233
8.17	Graphical representation of the relation among <i>LDA misclassification rates, average separation</i> and offset in the case MS20 applying the MAXDEV method . . . . .	234
8.18	Illustration of the mean-shifting procedure in the case MS3 with S500 and offset 7.39 (MAXDEV) . . . . .	237
8.19	Visualisation of the LDA boundaries for the two artificial data sets in the case MS3 (MAXDEV) . . . . .	239
8.20	Graphical representation of the relation among <i>LDA misclassification rates, average separation</i> and offset in the case MS3 applying the MAXDEV method . . . . .	240
8.21	Illustration of the mean-shifting procedure in the case MS1 with S500 and offset 40.45 (MAXDEV) . . . . .	242
8.22	Visualisation of the LDA boundaries for the two artificial data sets in the case MS1 (MAXDEV) . . . . .	243
8.23	Graphical representation of the relation among <i>LDA misclassification rates, average separation</i> and offsets in the case MS1 applying the MAXDEV method . . . . .	244
8.24	Illustration of the mean-shifting procedure for MINDEV in all MS cases with S500 . . . . .	248
8.25	Illustration of the mean-shifting procedure for MAXMEAN in all MS cases with S500 . . . . .	251
<b>9</b>	<b>Conclusions and Further Work</b>	<b>258</b>
9.1	<i>Misclassification rate</i> vs offset for various MS cases with the UN-SCALED and LOG-TRANSFORMED data . . . . .	265
9.2	<i>Misclassification rate</i> vs offset for various MS cases with the ROW-SCALED and LOG-TRANSFORMED data . . . . .	266
	<b>Appendix B - Simulation Results</b>	<b>276</b>
B.1	Visualisation of the LDA boundaries for the two artificial data sets in all four MS cases (MINDEV) . . . . .	282
B.2	Graphical representation of the relation among <i>LDA misclassification rates, average separation</i> and offsets in the case MS120 for method MINDEV . . . . .	283
B.3	Visualisation of the relation among <i>LDA misclassification rates, average separation</i> and offsets in the case MS20 with MINDEV . . . . .	284
B.4	Visualisation of the relation among <i>LDA misclassification rates, average separation</i> and offsets in the case MS3 with MINDEV . . . . .	285
B.5	Visualisation of the relation among <i>LDA misclassification rates, average separation</i> and offsets in the case MS1 with MINDEV . . . . .	286



B.6	Visualisation of the LDA boundaries for the two artificial data sets in all four MS cases (MAXMEAN) . . . . .	296
B.7	Visualisation of the relation among <i>LDA misclassification rates</i> , <i>average separation</i> and offsets in the case MS120 with MAXMEAN	297
B.8	Visualisation of the relation among <i>LDA misclassification rates</i> , <i>average separation</i> and offsets in the case MS20 with MAXMEAN	298
B.9	Visualisation of the relation among <i>LDA misclassification rates</i> , <i>average separation</i> and offsets in the case MS3 with MAXMEAN .	299
B.10	Visualisation of the relation among <i>LDA misclassification rates</i> , <i>average separation</i> and offsets in the case MS1 with MAXMEAN .	300
<b>Appendix D - Vignette for the Simulation Algorithm</b>		<b>339</b>
D.1	Example of a .csv file containing the spectral information of the epilepsy patients. . . . .	340
D.2	Example of the patients clinical and spectra information as stored in an R object of class <code>epiData</code> . . . . .	340
D.3	Example of the mean, stdev, median and covariance matrix information as stored in an R object of class <code>epiData</code> . . . . .	341
D.4	An example of the use of R function <code>createDataClass()</code> to obtain an R object of class <code>epiData</code> . . . . .	343
D.5	An example of the use of R function <code>generateSet()</code> to create a reference and a test set. . . . .	344
D.6	Results of a simulation analysis with the use of the R function <code>simulateData()</code> for option "pca". . . . .	345
D.7	PCA scores plot with the superimposed LDA boundary for the analysis in Figure D.6. . . . .	346
D.8	Results of a simulation analysis with the use of the R function <code>simulateData()</code> for option "stat". . . . .	347
D.9	Example of the graphical output of R function <code>plotData()</code> for the simulation experiment in Figure D.6. . . . .	348
D.10	Example of the output of R function <code>runSimulation()</code> with arguments <code>plotTrue</code> set to TRUE and <code>multiple</code> to FALSE. . . . .	349
D.11	Example of the output of R function <code>runSimulation()</code> with arguments <code>plotTrue</code> set to TRUE and <code>multiple</code> to TRUE. . . . .	351
D.12	Example of the output of R function <code>runSimulation()</code> with arguments <code>plotTrue</code> set to FALSE. . . . .	352
D.13	Example of the output of R function <code>plotMeanShifting()</code> . . . . .	353
D.14	Example of the output of R function <code>plotBoundaries()</code> . . . . .	354
D.15	Example of obtaining an object of class <code>statData</code> . . . . .	354
D.16	Example of an object of class <code>statData</code> . . . . .	355
D.17	Example of the output of R function <code>plotSimStats()</code> . . . . .	356
D.18	Example of the structure of a <code>bdSet</code> object. . . . .	356
D.19	Example of the structure of a <code>pcData</code> object. . . . .	357



# List of Abbreviations

AED	Anti-Epileptic Drug
APCI	Atmospheric Pressure Chemical Ionisation
API	Atmospheric Pressure Ionisation
CE	Capillary Electrophoresis
CI	Chemical Ionisation
EI	Electron Impact Ionisation
ESI	Electro-Spray Ionisation
FI	Field Ionisation
FT	Fourier Transformations
FWHM	Full width at Half Maximum
GC	Gas Chromatography
HPLC	High Performance Liquid Chromatography
ICR	Ion Cyclotron Resonance
LC	Liquid Chromatography
LDA	Linear Discriminant Analysis
PLS-DA	Partial Least Squares Discriminant Analysis
MALDI	Matrix-Assisted Laser Desorption/Ionisation
MS	Mass Spectrometry
NMR	Nuclear Magnetic Resonance
PCA	Principal Components Analysis
PCR	Principal Components Regression
SEC	Size-Exclusion Chromatography
SIMS	Secondary Ion Mass Spectrometry Ionisation
SRM	Single Reaction Monitoring
TLC	Thin-Layer Chromatography
TOF	Time of Flight
UPLC	Ultra High Performance Liquid Chromatography

# Notation

$M+$	Positive molecular ion (radical cation)
$V$	Potential
$z$	Charge
$m$	Mass
$B$	Magnetic flux density
$r$	Radius
$\frac{z}{m}$	Mass to charge ratio
$D_a$	Dalton
$u$	Unified atomic mass unit
$ppm$	Parts per million
$\mu$	Magnetic moment
$h$	Planck's constant
$I$	Net spin of a nucleus
$\gamma$	Gyromagnetic ratio
$N_i$	Number of nuclei in a level $i$ spin state
$k$	Boltzmann's constant
$T$	Temperature
$K$	Kelvin's temperature scale
$\omega_o$	Larmor frequency
$F$	Component column vector ( $n$ -dimensional)
$X$	Observed variables' column vector ( $n$ -dimensional)
$\lambda_i$	Eigenvalue for component $f_i$
$\tilde{H}$	Normalised entropy of data
$\phi$	Gleason–Staelin statistic

# Part I

## Project and Data Description

# Introduction

Part I contains the necessary background information for the project. Such information includes the definition of metabonomics (as the data to be analysed is of this type), the definition of epilepsy and related syndromes (since the data comes from the blood serum of people with epilepsy), the analytical techniques which can be used to generate the metabonomics data from the blood serum of the patients, and the pre-processing and pre-treatment methods which can be applied to the generated by NMR Spectroscopy data, to prepare this data for further statistical analysis.

The generation of information about the genome, the scope of this thesis, descriptions of the -omics, as well as of metabolic profiling, fingerprinting and target analysis, is the subject of Chapter 1. More specifically, the relationship between the various fields of bionomics is depicted graphically and a brief description of genomics, proteomics and metabonomics are given in Section 1.3 and of toxicogenomics in Section 1.4. The advantages of using the metabonomics technology, as well as brief descriptions of the main approaches used for the analysis of metabolic networks and pathways are also stated in the same section. Finally, in Section 1.5 a short description of chemometrics is given.

The definition of epilepsy and of epileptic syndromes, as well as the types of epileptic seizures and some important facts about epilepsy are given in Chapter 2, Section 2.2. The problem is described in Section 2.3 whereas the data to be analysed (clinical and spectral information and important characteristics of the patients in the data set) are mentioned in Section 2.4.

In Chapter 3, the two most important (and commonly used) analytical chemical techniques for the generation of metabonomics are discussed. More specifically, Mass Spectrometry (MS) and Nuclear Magnetic Resonance (NMR) Spectroscopy are described in detail and a comparison between these two techniques is attempted. MS is covered in Section 3.2, which includes the theoretical background, history and a description of the main components of an MS instrument and of a mass spectrum it produces, as well as a list of applications of MS to metabonomics studies. NMR spectroscopy is covered in Section 3.3, which likewise in MS, it

includes the theoretical background of NMR, a description of the main components of an NMR spectrometer and of an NMR spectrum. A list of applications of NMR to metabonomics studies is also given in this Section. Finally, the main advantages and disadvantages of the two metabonomics analytical techniques are given in Section 3.4.

Chapter 4, which completes this part of the thesis, deals with pre-processing and pre-treatment methods that can be used in NMR spectra to enhance the quality and accuracy of the generated metabonomics data removing any irrelevant information such as signal noise (pre-processing) and also to prepare the data so that it is suitable for statistical analysis (pre-treatment). Pre-processing methods such as binning, baseline correction, deconvolution and smoothing are discussed in Section 4.3. Pre-treatment methods such as scaling (row and column) and transformations are discussed in Section 4.4.

# Chapter 1

## Generating Information About the Genome

### 1.1 Introduction

One of the most important achievements of researchers in biomedical sciences is the decoding of the gene sequences of various organisms, one of which is the human. In addition, there are now large databases of single gene variations. Although the evaluation of gene expression (transcriptomics) and protein level (proteomics) changes has been extended significantly in the last few years, there is still a lot of research to be done in these areas (Fiehn, 2001). As will be seen in this thesis, another technology can be used to improve the understanding of how the various biological processes work. This is termed *metabonomics*, and in general, is regarded as a better technology than those mentioned previously, as it provides important evidence of molecular markers for the diagnosis of diseases and the evaluation of beneficial or adverse drug effects (Lindon, 2004). Metabonomics relates to the gene and protein expressions as well as to the metabolism, and considers also any environmental and physiological variation factors which can influence any part of the molecule (Bollard et al., 2005b). This chapter covers the scope of this research, as well as descriptions of the main functional genomic levels (*transcriptome*, *proteome*, *metabolome*) and the technologies used to study the functional networks and pathways of these genomic levels (*transcriptomics*, *proteomics*, *metabonomics/metabolomics*), as well as the effects of environmental and physiological variation factors to functional genomics (*toxicogenomics*). The three main ways of analysing metabolic networks and pathways, metabolite *profiling*, *fingerprinting* and *target analysis*, as well as the use of multivariate statistical techniques to chemical and/or biological data such as metabonomics data (called

*chemometrics*), are also briefly discussed in this chapter.

## 1.2 Scope of the Thesis

The purpose of this research is to investigate statistical techniques which can be used in the analysis of metabonomics data. The metabolic profiling of blood serum with newly diagnosed epilepsy will be used as an example. More specifically, the aim is to assess the ability of various clustering techniques to discriminate between two groups of patients, responders and non-responders to AEDs, by exploring the metabolic profiles of blood serum of patients with epilepsy. This investigation, hopefully, will confirm whether these clustering techniques can be used to identify any natural groupings in data such as those consisting of metabolic profiles.

## 1.3 Bionomics

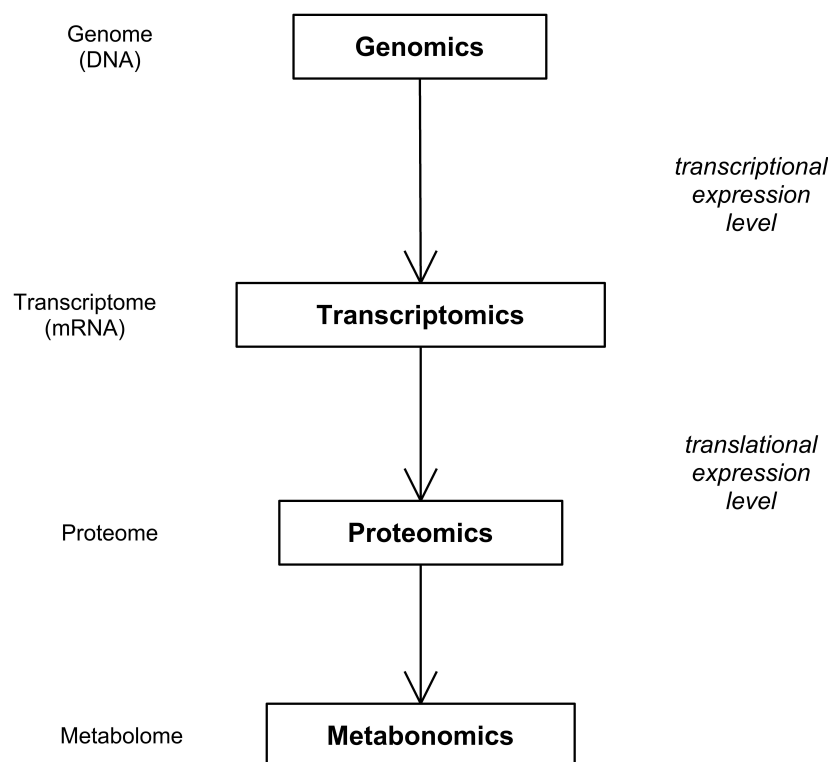
The main four "omics" technologies, also called as *bionomics*, are genomics, transcriptomics, proteomics and metabonomics (Lindon et al., 2001). The relationship between these technologies can be seen in Figure<sup>1</sup> 1.1.

### 1.3.1 Genomics/Transcriptomics

Genomics involve the study of an organism's entire genome. This field includes the observation and investigation of gene sequences and differences in those sequences between species and individuals, as well as of the variation of gene sequences in health and disease. More specifically, genomics study the differences in gene expression due to genetic modifications, diseases or toxicity, caused by compounds foreign to the organism, such as drugs (Lindon et al., 2001). This is a complex, lengthy and expensive approach and relative few organisms have been sequenced until now. However, the field of genomics cannot explain the biological consequences from changes to genes expression. For this reason, the field of proteomics has been developed.

---

<sup>1</sup>Source: Based partially in (Nielsen and Oliver, 2005), Figure 1 and (Oberemm et al., 2005), Figure 2.



**Figure 1.1:** Simplified Relationship between the main -omics technologies. There are also multiple feedback loops from metabolites to proteins and /or transcripts among others.

### 1.3.2 Proteomics

The study of the full set of proteins (the proteome) encoded by a genome. It involves the quantitative and qualitative measurement of the production of cellular proteins as a consequence of drug exposure and other pathophysiological processes (Lindon et al., 2001). There are many different approaches to address the very extensive range of proteins and most of them are based to some form of Mass Spectrometry. All proteomic measurements require a protein separation method such as 2D gel-electrophoresis. Proteomics are less expensive than genomics, but can be slow and labour-intensive. It is also very difficult to relate genomic and proteomic findings to known information about toxicity or toxicological endpoints (Lindon et al., 2000).

### 1.3.3 Metabonomics

While mRNA gene expression data and proteomic analyses cannot fully explain what actually happens in a cell, metabolic profiling can give an instantaneous picture of the physiology of the cell. Thus, the study of metabolic networks and pathways is required to complement the understanding of biological processes in



living organisms. Metabonomics is a technology which aims to achieve that goal. Lindon (2004) comprehensively defines metabonomics as:

*The quantitative measurement of the time-related multiparametric metabolic response of living systems to pathophysiological stimuli or genetic modification.*

More precisely, metabonomics can be defined as (Nicholson et al., 2007):

*The comprehensive and simultaneous systematic profiling of multiple meta-bolic levels and of their systematic and temporal changes caused by factors such as diet, lifestyle, environment, genetic effects and pharmaceutical effects, both beneficial and adverse, in whole organisms.*

This can be achieved by studying samples of various types such as biofluids (e.g. cerebrospinal fluid (CSF), blood plasma, blood serum, urine, seminal fluid, bile), tissue extracts (e.g. cardiac, liver, renal) and cell culture supernatants. Another technology, apparently similar to metabonomics, is termed *metabolomics*. Initially, involved mainly the study of in vitro systems in the plant science, but recently metabolomics have been used for the study of mammalian systems as well. Metabolomics involves the global analysis of all metabolites in a sample (Goodacre et al., 2004; Weckwerth and Morgenthal, 2005; Nielsen and Oliver, 2005; Griffin, 2004), whereas metabonomics is the analysis of metabolic responses to drugs or diseases (Lindon et al., 2004, 2006; Goodacre et al., 2004; Griffin, 2004; Lindon, 2004). Although metabolomics aims to identify and measure the dynamic set of all molecules present in an organism or biological sample, and metabonomics aims to identify target compounds and their biochemical transformations, both technologies now converge in methods and approaches used in the analysis of their data (Lindon et al., 2006; Weckwerth and Fiehn, 2002; Want et al., 2007; Ekins et al., 2005; Ryan and Robards, 2006).

### 1.3.4 Advantages of Metabonomics

According to Nicholson et al. (2007), the advantages of using metabonomics to biomedical applications can be summarised as:

- Using metabonomics it is easier to relate observed gene expression fold changes to conventional end-points (e.g in disease diagnosis and pharmaceutical evaluation) than transcriptomics.
- Gene expression and protein translation analyses are based almost exclusively in one analytical technique, mass spectrometry (MS), whereas meta-

bonomics is not restricted to MS.

- Metabonomics is faster, less labour-intensive and more technological advanced than proteomics.
- It involves the observation of the biochemical effects in an organism, thus representing a closer approach to real-world end-points than the other "omics" approaches.

### 1.3.5 Metabolic Profiling, Fingerprinting and Target Analysis

There are three main approaches which are consistently being used for the analysis of metabolic networks and pathways. These are *metabolite profiling*, *metabolite fingerprinting* and *metabolite target analysis* (Ryan and Robards, 2006; Nielsen and Oliver, 2005; Fiehn, 2002). Metabolite profiling is concerned with the identification and quantitation (by using a specific analytical technique), of a predefined group of known or unknown metabolites (e.g. a class of metabolites such as carbohydrates and amino acids), which belong to a selected metabolic pathway. This is the oldest and most established metabolite analysis approach and is considered as the precursor for metabonomics and metabolomics. Metabolite fingerprinting can be considered as spectra generated by analytical techniques such as NMR and MS, which provide a fingerprint of the metabolites produced by a cell. It aims to rapidly classify a large number of samples with the aid of multivariate statistics, without differentiation of individual metabolites or providing any information about specific metabolites. Metabolite target analysis, contrary to metabolite fingerprinting, aims to the qualitative and quantitative analysis of a specific metabolite or metabolites which participate in a specific part of the living system's metabolism. Thus, only signals from the required metabolites are retained for analysis, whereas the rest of the signals are being ignored.

### 1.3.6 Metabonomics Applications

Metabonomics can be applied to a wide range of applications. Especially, to mammalian systems, applications of metabonomics include the study of phenotypic and physiological effects (Bollard et al., 2005b), the pre-clinical drug candidate safety assessment (Lindon et al., 2003) and the disease diagnosis and therapeutic efficacy (Lindon et al., 2004). Among the various areas of application of metabonomics, an important area involves the investigation of multi-parametric

metabolic responses of mammalian systems to various diseases. More specifically, recently it has been possible to analyse biofluids, such as blood serum or urine, for the purpose of investigating the effects of drug administration to living organisms. In the case of human biofluid samples, metabonomics can be used to facilitate the diagnosis of diseases such as heart disease, cancer and epilepsy (Lindon et al., 1999). The investigation of the diagnosis of drug-resistant epilepsy and of possible insights in anti-epileptic drug-administration are examined in this thesis.

## 1.4 Toxicogenomics

Toxicogenomics is the study of how genomes respond to environmental stressors or toxicants. It combines genome-wide mRNA expression profiling (transcriptomics), cell and tissue-wide protein expression (proteomics), metabolite profiling (metabolites) and bioinformatics with conventional toxicology to understand the role of gene-environment interactions in disease and dysfunction (Oberemm et al., 2005; Schmidt, 2002). Toxicogenomics can also be used as a preventative measure for the prediction of adverse effects of drug treatment to living organisms. Diagnostic markers can be developed by correlating toxicogenomics studies to adverse toxicological effects in clinical trials. It is then theoretically possible to assess an individual's susceptibility to these adverse effects before administering a drug, so that the treatment of that individual can be done with a different drug, in case a marker of adverse effects is confirmed for this patient.

## 1.5 Chemometrics

Chemometrics, in general, involve the application of multivariate statistical techniques, pattern recognition methods and informatics to chemically-based data. The initial objective in metabonomics is to classify a spectrum (generated by a metabonomics analytical technique and containing e.g. the metabolic profile information of a patient) based on identification of its inherent patterns of peaks and secondly to identify those spectral features responsible for the classification. This approach can also be used for reducing the dimensionality of complex data sets, for example by two or three-dimensional mapping procedures to enable easy visualisation of any clustering or similarity of the various samples. In addition, *supervised* chemometric methods can be used to model multi-parametric data sets, so that the class of separate samples (a *validation* set) can be predicted

based on a series of mathematical models derived from the original data (the *training* set).

In the next chapter, the problem that this thesis is concerned with, will be discussed, as well as the data set that will be used for the statistical analyses. In addition, as the data concerns patients with epilepsy, some information about what epilepsy is and a few important facts about this disease (or disorder in some cases), are given.

# Chapter 2

## Project Description

### 2.1 Introduction

Despite the seminal advances in epilepsy research during the last century, it is still a considerable challenge for epilepsy researchers to fully understand the neurobiology of epilepsy. The complexity of this disorder is not only due to the fact that it involves among other living organisms the human, which is the most complex entity in the known world, but also due to the fact that many seemingly unrelated factors can affect significantly the levels of seizure activity in humans. Such factors include among others fever, sleep deprivation, hormonal disturbances, stress and drug treatment. This study aims to improve the understanding of the underlying mechanism of the response to AEDs treatment of patients with pharmaco-resistant epilepsy. That is, to study the effect of drug treatment in the reduction of seizure levels of epileptics. Therefore, the definitions of terms related to epilepsy such as epileptic disorder, epileptic seizure, epileptic syndrome and epileptic disease, as well as the main types of epileptic seizures and syndromes/diseases are stated in Section 2.2. A description of the problem under investigation is given in Section 2.3, whereas specific general clinical information about the patients participating in this research, the format of the data sets and the specific characteristics of the subjects in the original data set can be found in Section 2.4.

### 2.2 Definition of Epilepsy

To define the term epilepsy, one should be careful as there is no common agreement. The International League Against Epilepsy (ILAE) and the International Bureau for Epilepsy (IBE), define the term **Epileptic Disorder** as a chronic

neurologic condition characterized by recurrent epileptic seizures (Blume et al., 2001). They also give the definition of the term **Epilepsies** as those conditions which involve chronic recurrent epileptic seizures that can be considered epileptic disorders. Although many different opinions on these terms and definitions exist among the community involved in this matter (physicians, educators, researchers and others), a more complete way of defining the terms **epileptic seizure** and **epilepsy** are given below:

*An epileptic seizure is a transient occurrence of signs and or symptoms due to abnormal excessive or synchronous neuronal activity in the brain (Fisher et al., 2005).*

A description of epilepsy which involves this kind of seizure can be given as

*Epilepsy is a disorder of the brain characterized by an enduring predisposition to generate epileptic seizures and by the neurobiologic, cognitive, psychological and social consequences of this condition. The definition of epilepsy requires the occurrence of at least one epileptic seizure (Fisher et al., 2005).*

Furthermore, according to the latest ILAE classification of epilepsies, two different concepts have been proposed, namely **Epilepsy syndrome** and **Epilepsy disease** (Engel, 2006a). The former is defined as

*A complex of signs and symptoms that define a unique epilepsy condition with different etiologies, which must involve more than the seizure type*

whereas the latter is defined as

*A pathological condition with a single specific, well-defined etiology.*

It is important to note, that according to Fisher et al. (2005) there are three main elements which characterise epilepsy. First of all, at least one seizure is required to establish the presence of epilepsy. Secondly, under the above definition the diagnosis of epilepsy would also require an enduring disturbance of the brain, efficient to give rise to other seizures. Lastly but not less importantly, any neurobiologic, cognitive, psychological and social disturbances that some people with epilepsy appear to have should be assessed as part of their epileptic condition.

### 2.2.1 Types of Epileptic Seizures

A classification according to ILAE of the various epileptic seizures with respect to their clinical type is given below (Engel, 2006a,b; Devinsky, 1999):

### Self-limited Epileptic Seizures

- **Generalized seizures (convulsive or non-convulsive).** The main generalized seizures include types such as *tonic* and/or *clonic* seizures, *absences*, *myoclonic* seizure types, *epileptic spasms* and *atonic* seizures.
- **Focal seizures.** These involve in general *focal sensory* or *motor* seizures. The main categories of focal types include:
  - **Local or sensory.** *Neocortical* (with or without *local* spread) or *Hippocampal/Parahippocampal* seizures.
  - **Motor seizures.** These include *hyperkinetic* seizures and *dyscognitive* seizures with or without automatisms. The former affect neocortical areas whereas the latter limbic areas.
  - **Focal with contralateral spread to specific areas.** Such seizure types are among others, the *hemiclonic* seizures affecting the neocortical areas and the *gelastic* seizures affecting the limbic areas.
- **Neonatal seizures.** These are separated from the other self-limited epileptic seizures as they often display unique organizational features (Engel, 2006b).

### Continuous Seizures - Status Epilepticus

These seizure types include the various *generalized status epilepticus* types such as the *clonic*, *absence*, *tonic* and *myoclonic epilepticus* types and the *focal status epilepticus* types such as the *epilepsia partialis continua of Kojevnikov (EPC)*, the *aura continua*, the *limbic status epilepticus* and the *hemiclonic status*.

## 2.2.2 Epilepsies and Epileptic Syndromes

Similarly, epilepsies and epileptic syndromes can be classified according to ILAE with respect to the age of onset and related conditions, in the following way (Engel, 2006a,b):

1. **Neonatal period.** The neonatal epilepsies include the *Benign familial neonatal* seizures (BFNS), the *Early myoclonic encephalopathy* (EME) and the *Ohtahara syndrome*.
2. **Infancy.** These include among others the *West syndrome*, the *Myoclonic epilepsy in infancy* (MEI) and the *Dranvt* syndrome.
3. **Childhood.** Syndromes which belong to this category of epilepsies include among others the *Benign childhood epilepsy with centrotemporal spikes*

(BCECTS), the *Lennox-Gastaut* syndrome (LGS), the *Landau-Kleffner* syndrome (LKS) and the *Childhood absence* epilepsy (CAE).

4. **Adolescence.** The *Juvenile absence* epilepsy (JAE), the *Juvenile myoclonic* epilepsy (JME) and the *Progressive myoclonus* epilepsies (PME) are types of adolescence epileptic syndromes.
5. **Non-specific age relationship.** These syndromes include the *Autosomal - dominant nocturnal frontal lobe* epilepsy (ADNFLE), the *Rasmussen* syndrome, the *Familial temporal lobe* epilepsies and the *Gelastic seizures with hypothalamic hamartoma*.
6. **Special epilepsy conditions.** In this category of epilepsies, belong syndromes such as the *Reflex* epilepsies, the *Febrile seizures plus* (FS+) and the *Familial focal epilepsy with variable foci*.

### 2.2.3 Epidemiology of Epilepsy

Recent studies (Kwan and Brodie, 2000a; Loscher, 2002) show that epilepsy affects just under 1% of the population worldwide and about 4% of individuals over their lifetime (Loscher and Schmidt, 2002). More specifically, in Europe<sup>1</sup>, the estimated prevalence of epilepsy in 2004, was approximately 4.3-7.8 per thousand of individuals. The estimated total cost of epilepsy in Europe in 2004 was nearly 11 billion pounds (Pugliatti et al., 2007). In the USA, the disorder affects approximately 0.6% of the population and has a lifetime prevalence of nearly 3% (Devinsky, 1999).

Although in the last hundred years there have been many advances in achieving the goal of freeing epileptic patients from seizures and their side effects (Lowenstein, 2008), there are still many aspects of the disorder which have not been understood. There are no tools fully capable of the analysis of complex biological networks such as those related to seizures. In addition, seizures result from stochastic processes and the understanding of the underlying mechanisms that contribute to the reduction of the seizure levels in humans is still fairly basic.

## 2.3 Description of the Problem

Various studies have shown that more than 30 percent of patients with epilepsy cannot control adequately their seizures with drug therapy (Kwan and Brodie,

---

<sup>1</sup>In this case the 25 European Union member countries as well as Iceland, Norway and Switzerland



2000a,b; Devinsky, 1999). The latest evidence (Kwan and Brodie, 2000a,b; Hitiris et al., 2007) shows that there exist two different patient groups:

- The responder group, which includes those patients who show significant improvement by the use of a modest dose of one of the two available anti-epileptic drugs (AED).
- The non-responder group, which includes those patients who do not show any improvement (or any relief) from seizures, despite receiving an appropriate drug treatment.

Recent studies support the fact that the non-responder group represents more than 30 percent of all epilepsy cases (Kwan and Brodie, 2000a,b; Hitiris et al., 2007). Furthermore, other studies (Devinsky, 1999) indicate that drug resistant epilepsy is related to significant physical and social disability, and poor quality of life, as well as an increased risk of sudden, unexpected death. Therefore, the identification of good and/or poor prognosis markers is necessary if we want to find new ways to treat epilepsy, and to improve the timely administration of alternative treatment options, helping in this way to eliminate the dangerous consequences of uncontrolled seizures.

## 2.4 Data Description

Participants of this study are newly diagnosed epilepsy patients at the Epilepsy Unit, Western Infirmary, in Glasgow. Initially, serum samples were collected from 125 subjects during a period between February 2004 and March 2006. All patients have been treated with AEDs (Zweiri et al., 2010). The data gathered from these samples includes six months of clinical follow-up of the subjects.

### 2.4.1 Clinical Information of the Patients

The clinical information that was collected during these six months includes the following details:

- Personal information for each subject, such as gender, date of birth, weight, height and BMI
- Date of collection of the sample
- Date of subject's most recent seizures
- Date of acquisition of the NMR data for each sample
- Date of clinical review for each subject

- Type of seizures each subject had. The seizures were categorised in three categories:

**LRE** Localisation-related epilepsy

**IGE** Idiopathic generalised epilepsy

**UNC** Unclassifiable epilepsy

- Type of response to drug treatment (type of epileptic seizures after six months). There were 8 different types of response observed, numbered from 1 to 8, which afterwards were simplified into only three:

1 Improvement

2 No improvement

3 Unclassified.

An example of the information given for a subject can be seen in Figure 2.1.

Code	Initials	Gender	Date.Birth	Weight	Height	BMI
MND5-005	I-N	Male	09-Jul-80	69.9	175	23
MND5-036	S-C	Male	29-Dec-04	86.2	179	26.9
MND5-059	J-O	Female	29-Aug-86	60.3	169	21.1
MND5-066	M-W	Female	26-Jul-39	58.8	166	21.3
MND5-114	W-F	Male	26-Feb-57	62	167	22.2

Sample.date	Most.recent.seizure	NMR.acquisition	Clinical.review	Seizure.type	Simple.Sz.type
17-Mar-04	15-Nov-03	22-Sep-05	05-Jun-06	IGE	IGE
06-May-04	22-Apr-04	03-Nov-05	05-Jun-06	LRE	LRE
06-Sep-04	20-Aug-04	23-Nov-05	12-Jun-06	LRE	LRE
07-Oct-04	23-Sep-04	23-Nov-05	12-Jun-06	LRE	LRE
16-Nov-05	22-Sep-05	18-May-06	15-Nov-06	UNC	LRE

Out.6.w	Out.6.w.s	Out.6.m	Out.6.m.s	Out.12.m	Out.12.m.s
1	1	1	1	1	1
1	1	3	2	8	3
3	2	1	1	1	1
2	2	3	2	2	2
1	1	2	2	1	1

LRE: Localisation-related epilepsy  
IGE: Idiopathic generalised epilepsy  
UNC: Unclassifiable epilepsy

**Figure 2.1:** Clinical data for five of the subjects - cases 5,36,59,66 and 114. The *Out...s* fields are the simplified values for the corresponding fields, e.g. *Out.6.m.s* is the simplified response information after six months of follow-up of the patients (simplified from 8 categories of response to 3 categories, responders, non-responders and unclassified patients).

## 2.4.2 The Data Set

Information about the concentration of various metabolites (possible biomarkers) in the blood serum of each patient, is extracted after the application of the NMR process to the collected samples. Due to the very large number of metabolites that were observed (a few thousand different metabolites), a reduction in their number

was deemed necessary. Therefore during the preprocessing stage of the analysis, binning (bucketing) was applied. Bins of size 0.04 ppm (parts per million) were used in this project, reducing effectively the number of metabolites to 332 without losing important information. Part of such a data set can be seen in Figure 2.2, in which the first column contains the codenames for each of the 125 subjects, while the other columns are the bins of specific amounts in ppm. Each row contains the

	10.98000"	10.94000"	10.90000"	10.86000"	10.82000"	10.78000"	10.74000"	10.70000"
"MN05_1_1	19765905.78	20857921.46	22520901.7	23834579	24473946.07	24661607.43	25868051.26	26223592.22
"MN05_2_1	103762893.9	111267966.1	113939402.9	122659663.3	126323958	129476369.8	135206695.7	139108366.8
"MN05_3_1	31664477.26	32822319.93	35378616.23	37970914.05	39411752.87	41685273.47	41112976.97	42119613.46
"MN05_4_1	66805042.12	70705922.84	75489861.7	77524997.97	80597454.24	82905892.28	84012268.53	92274168.8
"MN05_5_1	57528704.2	60189215.23	62229856.13	64754370.21	65301680.77	68238290.16	72780635.54	74937475.01
"MN05_6_1	61874572.69	66223023.95	68704433.88	71281206.92	76481423.97	77258942.04	80582667.93	81617998.41
"MN05_7_1	115229258.2	12106097.93	119757890.4	125714739.1	127293099.6	131548437.9	143796339.8	138881496.6
"MN05_8_1	72082381.31	74373469.6	78446036.99	82640923.34	88533097.64	90785636.05	93030406.19	92858232.93
"MN05_9_1	35831223.66	36033895.97	38640510.99	42047008.83	43887840.35	44185234.56	45891561.6	46094060.35
"MN05_10_1	68164394.81	71555101.83	72727910.11	79669977.03	83876938.9	85512664.21	87693782.09	89865971.61
"MN05_11_1	31804285.76	34050353.12	36629392.52	38918835.98	39788771.67	39217365.36	39646992.86	40448673.94
"MN05_12_1	57190195.77	59311748.66	63041760.51	67120262.96	68601335.51	71290011.4	72753214.43	76781816.16
"MN05_13_1	118623748.2	125105449.1	129908736.7	130981468.4	135844462.9	137347896.5	143409093.6	148084516.8
"MN05_14_1	60280343.15	57742111.13	63551874.67	65248099.08	65268334.57	6492594.25	69917202.54	72627776.03
"MN05_15_1	121396893.9	125998215.5	131648174.7	138705403.9	141446315.1	146033127.7	152371801.9	153936986.4
"MN05_16_1	55697024.08	57329539.88	62725546.6	63164244.71	67200888.6	68026112.1	69721436.54	72118143.24
"MN05_17_1	147382368.8	155992806.7	160480980.1	17799147.43	182938113	187701291.7	190990907.7	201307030.4
"MN05_18_1	66591766.57	69662905.21	74098920.1	77166229.66	77749836.79	79763326.32	83339728.91	86530910.86
"MN05_19_1	63617656.54	66692438.11	70579454.76	75485768.85	81255006.18	81151612.34	82136236.68	84257338.83
"MN05_20_1	32266398.07	32747875.57	34410200.44	3555524.71	37747942.08	38611776.84	38725372.434	40415008.91
"MN05_21_1	131996455.3	142994821.7	14929677.7	152072490.8	157908879.5	163212490.6	168146987	176170997.9
"MN05_22_1	115977660.5	121015720.1	123564443.7	123171740.2	133207931.1	135258110.2	144509927.1	146980430.4

Figure 2.2: An example of a dataset for bins from 10.98 - 10.70

concentrations of the 332 metabolites for each patient. For illustrative purposes, in Figure 2.2 only bins of metabolites with chemical shifts in the range 10.98 - 10.70 are included for the first 22 patients, as the full data set is a matrix of dimensions 125 x 332. Three of the samples have to be removed, as these are known (for medical reasons) to be potential outliers, due to specific indications in their clinical data<sup>2</sup>. Therefore, after removing patients 23, 85 and 86 from the data set, there are 122 patients remaining. In addition, to identify any differences in the metabolites' levels between patients with and without response to AEDs treatment after six months follow-up, a number of patients whose response to the AEDs could not be classified, were removed temporarily from the data. In Table 2.1 can be seen the seizure types of the patients as they were diagnosed before and after simplification of the types. The remaining 25 patients will be used as

Table 2.1: Seizure type of patients in the current data set before and after simplification

Seizure Type	Simplification	
	Before	After
LRE	63	75
IGE	14	22
UNC	20	

testing data for the validation of the classification quality whenever and in case

<sup>2</sup>After personal communication with Dr. John Parkinson, Department of Pure and Applied Chemistry, University of Strathclyde, who generated the metabolomics data by NMR Spectroscopy of the blood serum samples of the patients.

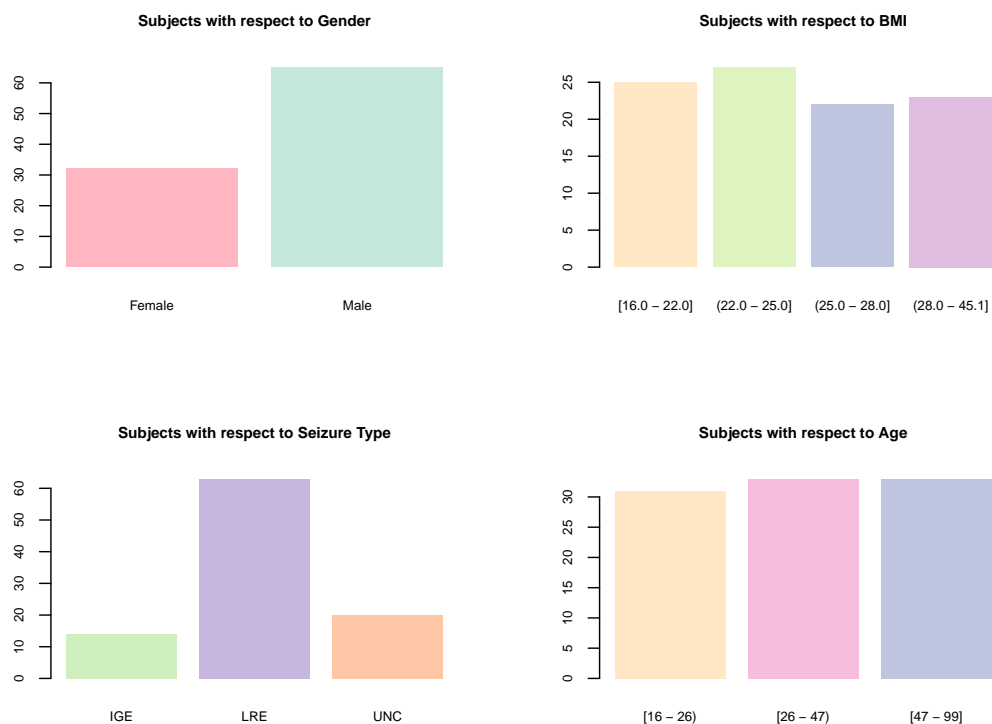
such data will be required by the various statistical analyses applied. Table 2.2 details the number of samples with respect to gender, seizure type and response to AEDs outcome.

**Table 2.2:** Clinical characteristics of the patients in the reduced data set

	Seizure type		AEDs Response	
	LRE	IGE	Responder	Non-Responder
Females	27	5	16	16
Males	48	17	36	29
Totals	75	22	52	45

### 2.4.3 Characteristics of Subjects in the Current Data Set

The most important features of the patients in the current data set can be seen in Figure 2.3. There are 65 men and 32 women in the reduced data set. The age



**Figure 2.3:** Graphical representation of the most important characteristics of patients in the current data set. The y-axis values in all plots are the number of patients in the current data set with regards to a clinical characteristic.

range of the patients is 17-99 years. Their Body-Mass-Index values were in the

range 16-45.1 (men: 16-45.1, women: 16.6-36). The patients were categorized according to their age into three groups of approximately the same size. These groups are [16,26], (26,47] and (47,99] with sizes 31, 33 and 33 respectively and recoding labels *Young*, *Middle* and *Old* respectively. In addition, the BMI values were recoded into four groups, namely *Small*, *Medium*, *Large* and *Huge* representing the intervals [16,22], (22,25], (25,28] and (28,45.1] respectively. The BMI group sizes are 25, 27, 22 and 23 respectively. With regards to the patients' seizure type, Table 2.1 gives the corresponding information for the current data set. From the post-treatment serum samples, and after six months of follow-up of the progress of patients, positive response to the drug treatment was observed in 52 patients (improvement of seizures or reduction of their occurrence), whereas 45 patients showed negative or no response to the drug treatment. Apart from these, there were another 25 patients whose response could not be considered or confirmed as either improvement or no improvement to the drug treatment, as mentioned previously.

In the next chapter, the most important aspects of the two most commonly used analytical chemical techniques, MS and NMR, for the generation of metabolomics data, are covered, including mentioning applications of these techniques to metabolomics studies.

# Chapter 3

## Metabonomics - Analytical Techniques

### 3.1 Introduction

A variety of analytical techniques for the generation of metabonomics data sets exist, each with its own advantages and disadvantages. The selection of the appropriate technique usually depends on the context of the investigation to be done e.g. plants, microbiological, mammals (Griffin, 2004; Weckwerth and Morgenthal, 2005), as well as the type of samples that are to be used for the analysis e.g. blood serum, urine, tissues, cerebrospinal fluid (Lindon et al., 2004, 2006, 2000, 1999; Goodacre et al., 2004). Common analytical techniques include *Mass spectrometry (MS)*, *Nuclear Magnetic Resonance (NMR) spectroscopy*, *Fourier Transform Infrared (FT-IR) spectroscopy* and *Ultra-Violet (UV and UV-vis) spectroscopy*.

Usually the type of samples used in the analysis, dictates the appropriate analytical technique to be used for the generation of the metabonomics data. In this context, MS is more suitable for tissues samples, whereas NMR is common practice to be used when biofluids are involved in the analysis. FT-IR spectroscopy is not used very often in metabonomics, as the main disadvantage of this analytical method is that it provides very poor distinction between the various classes of metabolites (Griffin, 2004; Lindon et al., 2006). Also, UV spectroscopy is used mainly to study the metabolic profiles of plants and plant materials (Bouchereau et al., 2000). Mass spectrometry requires a separation of the metabolic components before the actual MS analysis takes place using one of the many available separation techniques such as *gas chromatography (GC)*, *liquid chromatography (LC)*, *high performance liquid chromatography (HPLC)*, *ultra performance liquid chromatography (UPLC)* and *capillary electrophoresis (CE)*.

As every analytical technique has advantages and disadvantages when applied to metabonomics studies, new techniques have been developed which either connect MS and NMR in on-line hyphenated systems such as the HPLC-diode-array detector (DAD) mass spectrometry (MS) solid phase extraction (SPE)-NMR spectroscopy (HPLC-DAD-MS-SPE-NMR) hyphenated technique (Tang et al., 2009) or combine MS and NMR by applying both analytical platforms in biological samples to detect all possible metabolites, such as the combination of high-resolution magic angle spinning NMR (HR-MAS NMR) and GC-MS in the case of the identification of biomarkers in patients with colorectal cancer (Chan et al., 2009). These new techniques have been proved to be far more effective in identifying unknown compounds in complex biological samples than applying a single analytical technique. All these techniques generate complex multivariate data sets which need further analysis and interpretation with the appropriate chemometric tools.

In this chapter, a description of the main aspects of the two most important in metabonomics analytical techniques, MS and NMR, are given in Sections 3.2 and 3.3 respectively, as well as a comparison of these techniques in Section 3.4.

## 3.2 Mass Spectrometry (MS)

### 3.2.1 Definition

Mass spectrometry can be defined according to John B. Fenn, the 2002 Nobel Laureate in Chemistry and one of the most important contributors in MS, as (Siuzdak and Trauger, 2007):

*Mass spectrometry is the art of measuring atoms and molecules to determine their molecular weight. Such mass or weight information is sometimes sufficient, frequently necessary, and always useful in determining the identity of species. To practice this art one puts charge on the molecules of interest, i.e. the analyte, then measures how the trajectories of the resulting ions respond in vacuum to various combinations of electric and magnetic fields. Clearly the sine qua non of such a method is the conversion of neutral analyte molecules into ions. For small and simple species the ionisation is readily carried by gas-phase encounters between the neutral molecules and electrons, photons, or other ions. In recent years, the efforts of many investigators have led*

*to new techniques for producing ions of species too large and complex to be vaporised without substantial, even catastrophic, decomposition.*

Mass spectrometry is used as a tool for measuring the molecular mass of a sample. Information concerning the chemical structure of the mass can be generated using instruments called *mass spectrometers*. These instruments are often used for industrial and academic research purposes. A mass spectrometer creates charged particles from molecules. These particles are then analysed in order to provide information about the molecular weight of the mass and its chemical structure. Mass spectrometry can be applied in many areas, such as biotechnology, pharmaceutical, clinical, environmental and geological applications. It can also assist in metabolome analysis, creating spectra that provide a fingerprint of the metabolites that are produced by a cell (metabolite fingerprinting) and allowing the analysis of a group of specific metabolites e.g. a class of metabolites such as sulfides, hormones and vitamins.

### 3.2.2 Theoretical Background of MS

Mass Spectrometry is a technique which requires the use of charged molecules in order to generate data for a compound of interest. More specifically, after the sample has been introduced, it is necessary to convert the neutral molecules into ions. Ionisation of neutral molecules means to charge positively or negatively the molecules using one of the many available methods, thus obtaining molecular ions and other fragments. Methods to ionise molecules include, among others, the addition or subtraction of protons (called protonation and deprotonation respectively), and the ejection or absorption of an electron in the molecule of interest, known as electron ejection (Siuzdak and Trauger, 2007; Van Bramer, 1998). The physical state of the molecule and the amount of ionisation energy, are two important things that often determine the ionisation method to be used. The usual ionisation method used in MS is to excite (often with electron beams) the neutral molecule, forcing it to eject an electron, producing a positive molecular ion (radical cation  $M^+$ ) and possibly other ion fragments. The whole ionisation process occurs inside mechanical devices called ionisation sources (ionisers). Before exiting the source, the ions are exposed to an electric field of fixed voltage (potential)  $V$ , causing them to accelerate their exit towards another sector (magnetic sector-analyser) with potential energy  $zV$ . During the acceleration process, that potential energy is completely converted into kinetic energy (Duckett and Gilbert, 2002). The relation between potential and kinetic energy of such an accelerated ion of mass  $m$ , charge  $z$  and acquired velocity  $v$ , is given by the relation (3.2.1) below:

$$zV = \frac{1}{2}mv^2. \quad (3.2.1)$$



Upon reaching the magnetic sector, the ions are subjected to a magnetic field of magnetic flux density  $B$ . This forces them to follow a curved trajectory of radius  $r$  and the magnitude of this force is given by  $Bzv$ . The relation between this force and the movement of the ions is given by equation (3.2.2):

$$Bzv = \frac{mv^2}{r}. \quad (3.2.2)$$

From equations (3.2.1) and (3.2.2) an expression (3.2.3) relating the mass to charge ratio of the ions to the applied magnetic field can be found:

$$\frac{m}{z} = \frac{B^2 r^2}{2V}. \quad (3.2.3)$$

From equation (3.2.3), it can be seen that it is possible to select those ions which reach the detection stage of the procedure, by just varying density  $B$  of the magnetic field (Brisdon, 2003). As the ions approach a device called a detector, their signal is recorded and since they have different mass to charge ratios, a spectrum of the various ion signals is produced. This is in fact, a plot of the number of ions detected versus their mass to charge ratio,  $\frac{m}{z}$ , and is called a mass spectrum.

### 3.2.3 History of MS

The first step in the development of Mass spectrometry took place in 1897, when Sir J. J. Thompson studied the phenomenon of electrical discharges in gases, at the University of Cambridge. These studies resulted in the discovery of the electron. He also constructed, during the first decade of the 20th century, the first mass spectrometer for the purposes of determining the mass-to-charge ratios of ions. An improved version of a mass spectrometer to allow the study of isotopes was designed by F. W. Aston shortly after the First World War. At about the same time, A. J. Dempster developed the first electron impact source, which was used for the ionisation of volatilised molecules.

Four different contributions took place during the period between 1946 and 1953 (Borman et al., 2003). W. E. Stephens, at the University of Pennsylvania, introduced the concept of Time of Flight (TOF) MS in 1946. A TOF analyser is used for the determination of large biomolecules' mass as it has almost limitless mass range. In 1949, Hipple, Sommer and Thomas formulated the idea of Ion Cyclotron Resonance (ICR) which allows the detection of ions sequentially. M. B. Comisarow and A. G. Marshall combined ICR with Fourier Transformations (FT) to develop FT-ICR MS. This technique made possible the measurement of many different ions at once. In 1953, Nier and Johnson developed the double-focusing instrument to make possible the analysis of isotopes. In the same year, Paul and Steinwedel introduced the quadropole mass analyser, which had a great dynamic range and good stability, making it especially

suitable for quantitative analysis and drug discovery applications.

In the field of molecular analysis, there were two important developments. One of them was the Electrospray ionisation (ESI) technique, which was described by M. Dole in 1968. Despite this fact, it was J. B. Fenn who applied this technique for the first time, in 1984, in biomolecular analysis. The other development was the Matrix-assisted laser desorption/ionisation technique (MALDI). It was introduced in 1983 by two different research groups, i.e. K. Tanaka at Shimadzu Corp. and F. Hillenkamp and M. Karas at the University of Frankfurt. Figure<sup>1</sup> 3.1 illustrates the most important achievements in the development of Mass Spectrometry during the last hundred years.

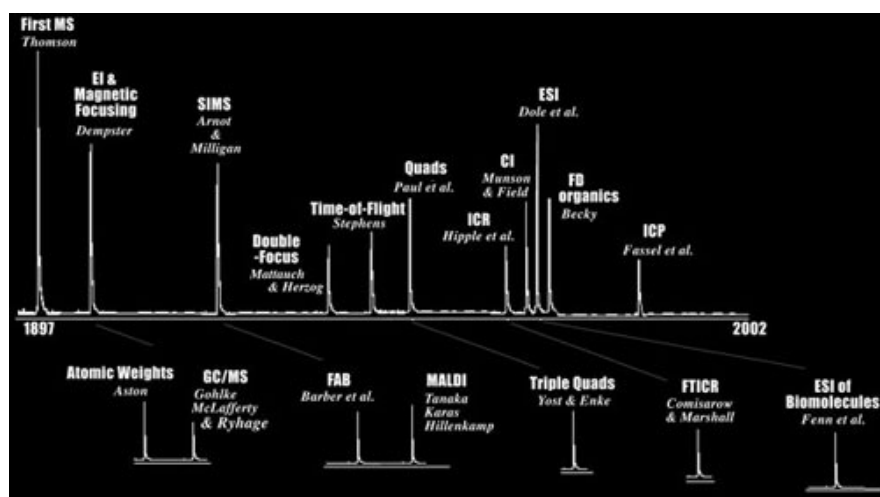


Figure 3.1: Important timelines and contributions to Mass Spectrometry

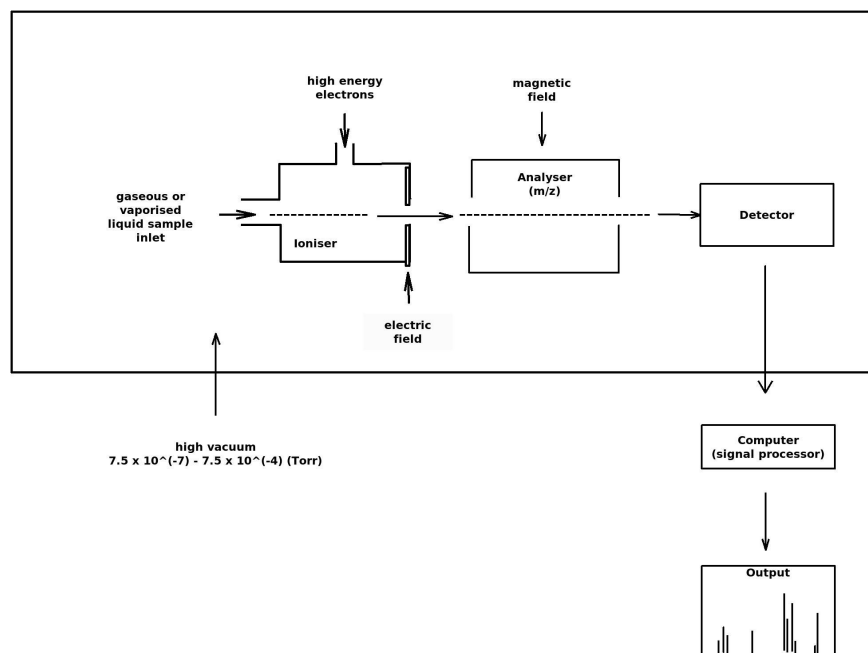
The further development of the techniques used in MS during the last years of the 20<sup>th</sup> century directed the research towards pharmacokinetics, which involves small molecule drug analysis and protein identification with the use of peptide mass mapping. More recently, MS has been applied to clinical studies, as a rapid and cheap neonatal screen for more than 30 different diseases. The latest achievements include the use of MS instrumentation for the generation of intact viral ions of millions of Daltons (Da) size and for confirming the preservation of virus's structure and virulence (Borman et al., 2003).

### 3.2.4 Description of Mass Spectrometers

Mass spectrometers consist of three main components, namely the ioniser (or ionisation source), the ion analyser and the detector (Glish and Vachet, 2003). Initially, the sample must be inserted into the ioniser of the instrument. Having done that, the molecules of the sample are ionised, since it is easier to work with ions than neutral molecules. These ions are extracted into the ion analyser of the mass spectrometer, in

<sup>1</sup>Source: Scripps Centre for Metabolomics and Mass Spectrometry

order to achieve their separation according to their mass ( $m$ ) - to - charge ( $z$ ) ratios ( $\frac{m}{z}$ ). Finally, the separated ions are recorded in the detector part of the instrument, and the produced signal is sent to a data system with the  $\frac{m}{z}$  ratios stored together with their relative abundance, to be presented in the format of a  $\frac{m}{z}$  spectrum. To increase the chances of the ions travelling through the instrument without any obstruction by air molecules, the components of the mass spectrometer are usually maintained under high vacuum. Figure 3.2 depicts the main parts of a mass spectrometer. It should be



**Figure 3.2:** Schematic of the main components of a mass spectrometer

noted that the ionisation method in use and the type and complexity of the sample might affect the way in which the sample is introduced to the instrument. Therefore, the sample can be introduced to the ioniser directly, or it might be necessary to apply a type of chromatography during the travel of the sample through the ioniser. In the latter case, the instrument is coupled to a chromatography separation column, which causes the sample's separation into a number of components. These components enter the instrument sequentially for individual analysis. The three most common types of chromatography are:

- **Liquid Chromatography<sup>2</sup>(LC-MS)**, which can be applied on any kind of stationary phase e.g. reversed phase, normal phase or ion exchange, coupled with mass spectrometry (Williams and Fleming, 1995; Kealey and Haines, 2002). In this case, analytes are separated by their chemical properties such as hydrophobicity, hydrophilicity or charge. LC is usually used as a preparation for the purification and isolation of some components in a mixture. In the case of more

<sup>2</sup>Often is needed high performance liquid chromatography (HPLC-MS) or ultra high pressure liquid chromatography (UPLC-MS).

analytical separations of solutions for the purpose of detection or quantification, more sophisticated instruments are needed, such as HPLC or UPLC instruments. These provide higher resolutions and shorter amounts of time for the analyses. A diagram of a typical LC separation procedure can be seen in Figure<sup>3</sup> 3.3.

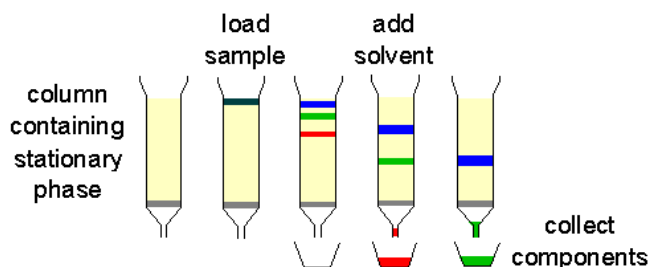


Figure 3.3: Liquid chromatography (LC) separation procedure

- **Gas Chromatography (GC-MS)**, which is usually applied to coated capillary columns, coupled with mass spectrometry (Williams and Fleming, 1995; Kealey and Haines, 2002). Here, analytes are separated by their boiling point and their interaction with the liquid layer covering the capillary in the gas phase. It is used mainly when the organic compounds to be separated are volatile. The main components of a gas chromatograph are a flowing mobile phase (usually an inert gas such as helium, argon or nitrogen), an injection port, a separation column with the stationary phase, a detector and a data recorder (Figure<sup>4</sup> 3.4).

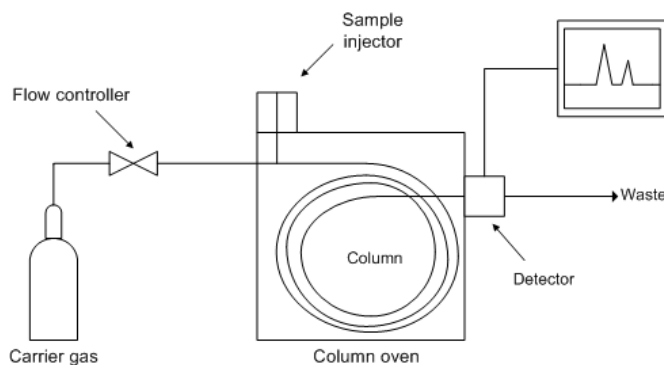


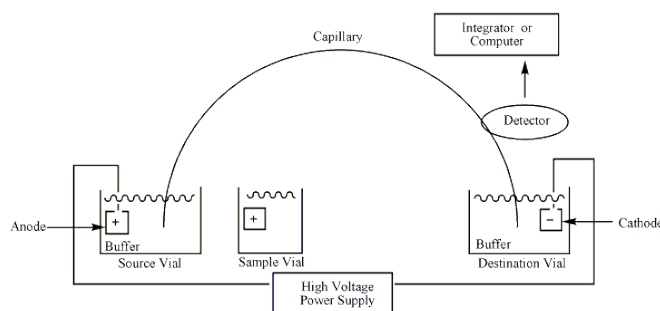
Figure 3.4: Diagram of a gas chromatography mass spectrometer (GC-MS)

- **Capillary Electrophoresis (CE-MS)** coupled with mass spectrometry (Kealey and Haines, 2002). In this type of chromatography, electrically charged analytes are separated by their mobility in a capillary filled with an electrolyte under the influence of an electric field (Figure<sup>5</sup> 3.5). The higher the electric field is, the more efficient the separation will be and the less time will be needed for the separation.

<sup>3</sup>Source: <http://www.chemistry.nmsu.edu/Instrumentation/lc-schem.gif>.

<sup>4</sup>Source: [http://upload.wikimedia.org/wikipedia/commons/8/87/Gas\\_chromatograph.png](http://upload.wikimedia.org/wikipedia/commons/8/87/Gas_chromatograph.png).

<sup>5</sup>Source: <http://upload.wikimedia.org/wikipedia/commons/9/99/Capillaryelectrophoresis.gif>.



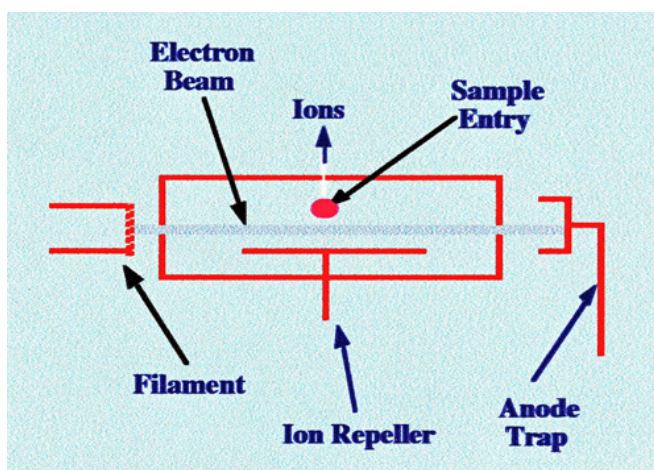
**Figure 3.5:** Diagram of a *capillary electrophoresis* mass spectrometer (CE-MS)

Other chromatographic methods that can be used for separation purposes are the *size-exclusion chromatography* (SEC) and the *thin-layer chromatography* (TLC) (Kealey and Haines, 2002).

### 3.2.4.1 Ionisation Process

Once the sample has entered the ioniser, there are a number of different methods that can be used for its ionisation (Siuzdak and Trauger, 2007). The type of sample and the mass spectrometer available are the main factors for choosing which ionisation method to use. The majority of biochemical analyses are done using the following three methods:

- **Electron Ionisation (EI)**. This is the standard ionisation method in mass spectrometry (Figure<sup>6</sup> 3.6). The sample is introduced into the Electron source (in-



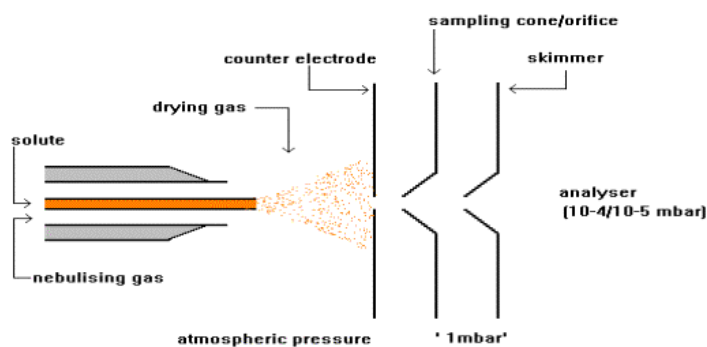
**Figure 3.6:** Basic diagram of an *electron* ionisation (EI) source

side a high vacuum) as a vapour, usually from a reservoir (in the case of gases and volatile liquids) or from a heated probe (for non-volatile liquids and solids). Sample molecules collide with high energy electrons (which a glowing filament produces). Ions are formed when the energy transferred exceeds the molecules'

<sup>6</sup>Source: [http://www.analyticalspectroscopy.net/ap8\\_html\\_m28d4b6e1.jpg](http://www.analyticalspectroscopy.net/ap8_html_m28d4b6e1.jpg).

ionisation energy. To send ions out of the source and to the mass analyser, an extraction voltage (500 - 10,000 V, depending on the type of instrument) needs to be applied. The efficiency of the ionisation is increased by placing the inner source between the poles of a small magnet, causing the electrons to travel with a helical trajectory. The Electron Ionisation method is especially useful for producing diagnostically useful fragment ions for structure elucidation, highly reproducible spectra and linear signal-response curves for use in quantitative analysis.

- **Electrospray Ionisation (ESI).** This is an ionisation method which belongs to the techniques called *Atmospheric Pressure Ionisation (API)* techniques. ESI is most suitable when the analysis involves polar molecules of molecular mass size from less than 100 Da up to more than 1,000,000 Da. In ESI, a fine spray of charged droplets is created by applying a high voltage (usually about 1-4 kV) to a capillary containing a flowing liquid. The use of a co-axial nebuliser gas, such as nitrogen, is often useful for improving the process (Figure<sup>7</sup> 3.7). ESI is suit-



**Figure 3.7:** Basic diagram of an *electrospray* ionisation (ESI) source

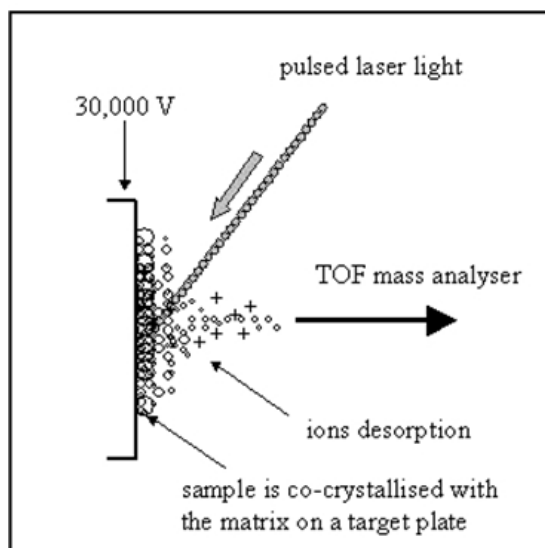
able for the analysis of organic compounds with medium - high polarity. Since positive ionisation is dependent on protonation, molecules containing basic functional groups work well in this mode. Negative ionisation, in contrast, functions by deprotonation, thus the presence of acidic functional groups is a prerequisite for reasonable limits of detection. Amino, amide, ester and aldehyde are some of the functional groups suitable for positive ESI, whereas functional groups such as carboxylate, phenol and imide are suitable for negative ESI.

- **Matrix Assisted Laser Desorption Ionisation (MALDI).** This is a method for laser desorption ionisation. The sample is mixed with a saturated solution of matrix<sup>8</sup> and a drop deposited on the MALDI target. After the solvent has been evaporated and the matrix been crystallised, the target is placed in the mass spectrometer source and is irradiated with pulses of laser light. A transfer of energy

<sup>7</sup>Source: [http://www.astbury.leeds.ac.uk/facil/MStut/mstutorial\\_files/image004.gif](http://www.astbury.leeds.ac.uk/facil/MStut/mstutorial_files/image004.gif).

<sup>8</sup>An organic compound with a strong absorption at the laser wavelength

between excited matrix molecules and sample molecules takes place, having as a result to desorb both from the condensed state. Once the molecules are in the vapour phase, there are proton transfers between matrix and sample which result in ion formation. Ions are then accelerated out of the source by applying a high potential (usually 20 kV) to a series of extraction electrodes and lenses (Figure<sup>9</sup> 3.8). This method is more suitable when the analysis involves thermolabile, non-



**Figure 3.8:** Basic diagram of a *matrix assisted laser desorption* ionisation (MALDI) source

volatile organic compounds and more specifically the high molecular mass ones. MALDI can be used in biochemical areas for the analysis of proteins, peptides, oligonucleotides and other compounds. MALDI is also very useful for characterising synthetic polymers, large organic molecules and organometallic complexes.

Other ionisation sources include among others, the *Atmospheric Pressure Chemical Ionisation* (APCI), the *Chemical Ionisation* (CI), the *Secondary Ion Mass Spectrometry Ionisation* (SIMS) and the *Field Ionisation* (FI) (Siuzdak and Trauger, 2007; Barwick et al., 2006). A comparison of the specifications of the most commonly used ionisation sources can be seen in Table E.1 (Source: [http://masspec.scripps.edu/mshistory/whatisms\\_details.php#Basics](http://masspec.scripps.edu/mshistory/whatisms_details.php#Basics)).

#### 3.2.4.2 Mass Analysis

After the extraction of the ions from the ioniser, they enter the ion analyser in order to be separated to their mass-to-charge ( $\frac{m}{z}$ ) ratios. The most commonly used mass analysers are (Siuzdak and Trauger, 2007; Van Bramer, 1998):

- **Single Quadrupole.** This is a mass-selective ion filter that consists of four parallel electronically operated metal rods. A varying voltage is applied, resulting

<sup>9</sup>Source: <http://www.chem.pitt.edu/sites/default/files/users/Bhg5/figure%205.jpg>.



in a fluctuating electric field. The potential applied to the rods consists of a DC ( $U$ ) and RF ( $V\cos\omega t$ ) component (Figure<sup>10</sup> 3.9). The main advantages of

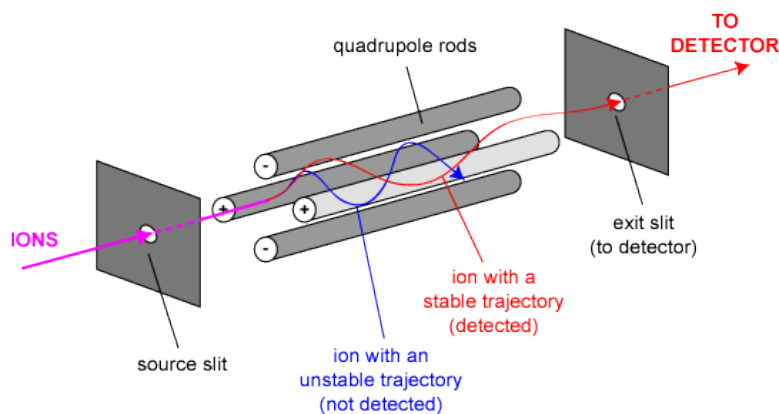


Figure 3.9: Single quadrupole mass analyser

quadrupole analysers are their robust and compact design.

- **Triple Quadrupole.** This analyser consists of three quadrupole devices coupled in a linear array. In the single reaction monitoring mode (SRM), the initial ions travel through the first quadrupole and then enter the second quadrupole device used as a collision cell in order to generate product ions. At the end, the last quadrupole filters the product ions according to their mass, obtaining the required ions. An example of a triple quadrupole mass analyser can be seen in Figure<sup>11</sup> 3.10.

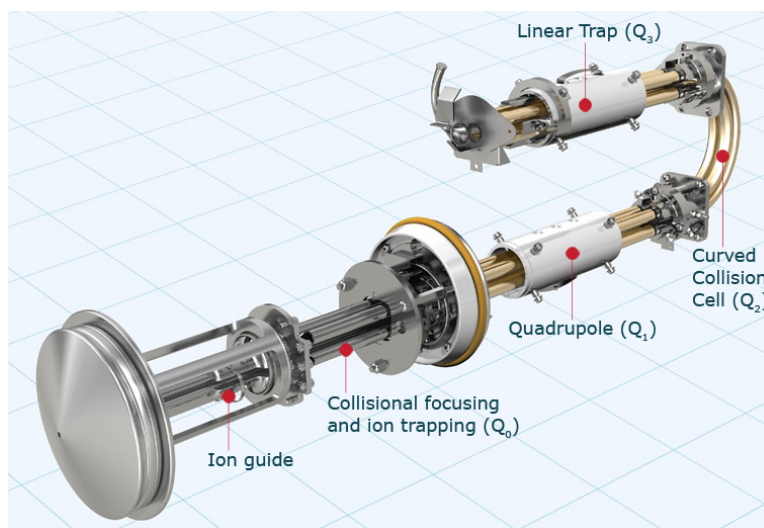


Figure 3.10: Triple quadrupole mass analyser

- **Iontraps.** This type of analyser initially forces ions into stable orbits and subsequently releases them, collecting and storing them according to their mass.

<sup>10</sup>Source: <http://www.chm.bris.ac.uk/ms/images/quad-schematic2.gif>.

<sup>11</sup>Source: <http://www.chromacademy.com/essential-guide/dec2011/figure-12.jpg>.



These ions too can be broken into other ions which can also be analysed. An example of an iontrap mass analyser can be seen in Figure<sup>12</sup> 3.11.

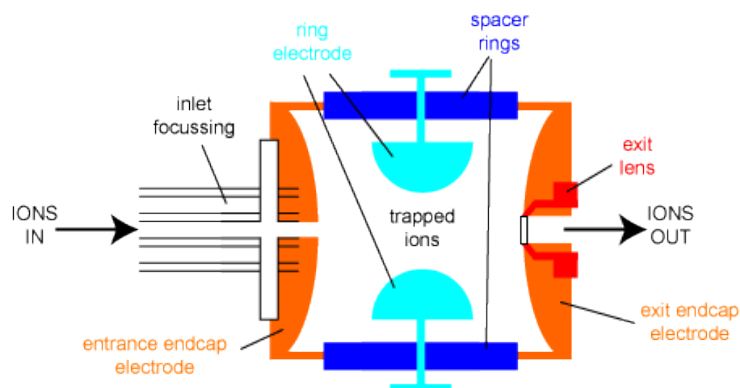


Figure 3.11: Iontrap mass analyser

- **Time-of-Flight (TOF)**. These analysers accelerate ions simultaneously, thus each ion obtains the same kinetic energy as any of the others. Consequently, the ions travelling through an evacuating flight tube with a fixed distance are subjected to separation according to their mass-to-charge ratio and velocity. The basic layout of a simple linear TOF analyser can be seen in Figure<sup>13</sup> 3.12. Their

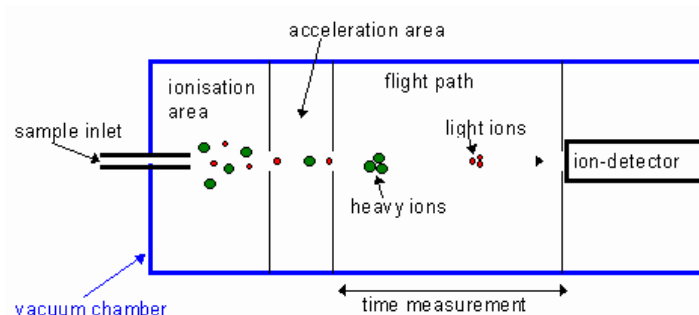


Figure 3.12: Time-of-Flight mass analyser

capability of high resolutions between 5000 and 20000 (FWHM), as well as their relatively small size and low cost are the main advantages of the TOF analysers.

Other types of analysers are the TOF Reflectron, the Fourier Transform MS and the Magnetic Sector (Siuzdak and Trauger, 2007; Van Bramer, 1998). The above-mentioned types of mass analysers differ in the  $\frac{m}{z}$  ranges that they can cover, mass accuracy and achievable resolution. Also, they are not always compatible with every ionisation method. Most of these types of analyser with their specific details (capabilities) can be seen in Table E.2 (Source: [http://masspec.scripps.edu/mshistory/whatisms\\_details.php#Basics](http://masspec.scripps.edu/mshistory/whatisms_details.php#Basics)).

<sup>12</sup>Source: <http://www.chm.bris.ac.uk/ms/images/iontrap-schematic.gif>.

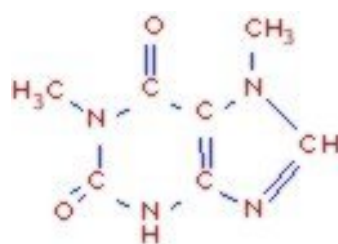
<sup>13</sup>Source: [http://www.kore.co.uk/graphics/MS-200\\_tof.gif](http://www.kore.co.uk/graphics/MS-200_tof.gif).

### 3.2.4.3 Detection of the Ions

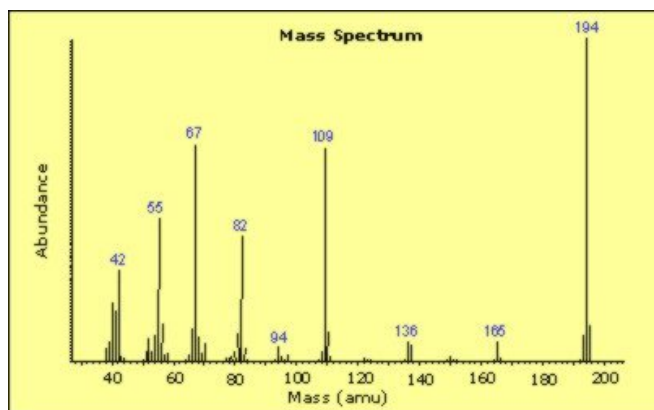
The last component of the mass spectrometer, the detector, monitors the ion current, amplifies it and the signal is afterwards transmitted to the data system, to be recorded as mass spectra. Detection of ions can be done in many different ways, depending on the type of mass spectrometer in use. The most commonly used detectors are the *Electron Multiplier*, the *Faraday Cup*, the *Photomultiplier Conversion Dynode* and the *Charge (or Inductive) Detector* (Siuzdak and Trauger, 2007). A comparison of the most commonly used detectors can be seen in Table E.3 (Source: [http://masspec.scripps.edu/mshistory/whatisms\\_details.php#Basics](http://masspec.scripps.edu/mshistory/whatisms_details.php#Basics)).

### 3.2.5 Mass Spectrum

In an MS plot, the  $\frac{m}{z}$  values of the ions can be seen against their intensities, thus providing information such as the number of components in the sample, the molecular mass of each of these components and also the relative abundance of the various components in the sample. An example of a  $\frac{m}{z}$  plot for a typical sample of an isolated compound ( $\sim 1$  nanogram) can be seen in Figure 3.13. This is a plot of relative intensity (abundance)



(a) Structural formula of a compound



(b) M/Z plot for the isolated compound (mass spectrum)

**Figure 3.13:** Example of a Mass Spectrum for a compound

versus the mass-to-charge ratio ( $\frac{m}{z}$ ). A number of peaks can be seen, of which the most

intense (most abundant) is characterised as the *base peak*. The rest of the peaks are noted with respect to the intensity of the base peak.

The peak having the highest molecular mass, of those observed in the spectrum, will usually be the parent molecule, termed as the molecular ion ( $M^+$ ). From the plot in Figure 3.13, it can be seen that the most abundant peak is at Mass 194 *u*. This peak represents the base peak, and due to the fact that it has the highest molecular mass (194) as well, it is also the molecular ion. More often than not, this is not the norm, as it usually is the case that the molecular peak or peaks in an MS plot differ from the most abundant one. The rest of the peaks in this plot are ion fragments of the initial neutral molecule, with various molecular ion masses.

The mass analyser of the spectrometer plays a very important role to the way that the spectra are represented. A mass spectrum depends on the *accuracy*, the *resolving power*, *mass range* and the *scan speed* of the analyser (Barwick et al., 2006; Webb et al., 2004).

The *accuracy* is the detail with which a mass analyser can provide  $\frac{m}{z}$  information and depends on the instrument's stability and resolution, e.g. a 1000 Da peptide to  $\pm 0.1$  Da (0.01%).

The *resolving power* is the ability of a mass spectrometer to distinguish between ions of different  $\frac{m}{z}$  ratios. The greater the resolution is, the better the ability to differentiate ions is. It is defined (Equation 3.2.4) as

$$Resolution = \frac{M}{\Delta M} \quad (3.2.4)$$

where  $M$  is the mass-to-charge ratio  $\frac{m}{z}$  and  $\Delta M$  is the full width at half maximum (FWHM). If the resolution is high enough, it could allow for the separation of an ion's individual isotopes (the narrowing of peaks allows the determination of an ion's position more accurately).

The *mass range* of an analyser is effectively its  $\frac{m}{z}$  range. Depending on the type of analyser (quadrupole, ion traps, TOF etc.), the  $\frac{m}{z}$  range will differ. For example, a typical scan range for a quadrupole analyser is up to  $\frac{m}{z} = 3000$ , whereas a TOF analyser has virtually unlimited  $\frac{m}{z}$  range.

The *scan speed* is the rate at which an analyser scans over a specific mass range. This usually takes a few seconds, but it depends on the type of analyser used. For example, a TOF analyser needs a few milliseconds or less to perform a complete analysis.

### 3.2.6 Applications of MS to Metabonomics Studies

Mass Spectrometry is an analytical technique which has been used extensively in metabolic fingerprinting and metabolite identification. Despite the fact that there is a large

number of MS studies on plants and plant extracts, as well as on model cell system extracts, during the last few years the number of applications of MS to mammalian studies has increased significantly (Lindon et al., 2006). Areas of applications include clinical applications such as the metabolic profiling of patients with colorectal cancer (Chan et al., 2009), xenobiotic toxicity assessments with respect to drug treatment (Idborg et al., 2005), to hepatotoxicity induced by  $CCl_4$  and dimethylnitrosamine (Lin et al., 2009; Sun et al., 2010) and to nephrotoxicity (Gamache et al., 2004). Other applications include physiological variation due to various factors such as gender differences (Dixon et al., 2007; Plumb et al., 2005), age differences (Plumb et al., 2005), strain differences (Wilson et al., 2005) and diurnal effects (Plumb et al., 2005).

## 3.3 Nuclear Magnetic Resonance Spectroscopy (NMR Spectroscopy)

### 3.3.1 Introduction

Nuclear magnetic resonance spectroscopy is one of the most important analytical techniques. It is used extensively in chemical applications, providing detailed information on molecular structure, both for pure compounds and in complex mixtures. NMR methods can also be used to study metabolite molecular dynamics and mobility, as well as substance concentrations (Lindon et al., 2006). NMR spectroscopy can be applied in a vast array of different types of samples either biofluids, cells or tissues. Especially as far as biofluid samples are concerned, NMR studies can be divided into two main categories: *analytical* and *dynamic*. The former involve the collection and quantitation of NMR spectra and their interpretation, whereas the latter involve the detailed understanding of the interactions of the components in the whole biological matrix of the organism. Dynamic NMR analyses include enzymatic biotransformations, metal complexation reactions, binding of small molecules to macromolecules, and cellular compartmentation. These can occur to varying degrees in many different biofluids (Lindon et al., 1999). Although NMR was initially used as a tool for molecular structural elucidation, it has proved to be very successful in characterising the functional effects of diseases, toxicity, physiological variation and genetic modification, in living organisms' biochemical profiles.

### 3.3.2 Theoretical Background of NMR

NMR is the phenomenon which occurs when the nuclei of specific atoms, while held in a static magnetic field, are exposed to a second oscillating magnetic field. This phenomenon depends on whether the nuclei possess a property called spin. As NMR

is concerned with matter, spectroscopy is the study of the interaction of matter with electromagnetic radiation. Therefore, NMR spectroscopy is the use of the NMR phenomenon in order to examine the various physical, chemical and biological properties of matter.

More specifically, NMR spectroscopy is a non-destructive technique that provides detailed information about the molecular structure of pure compounds and complex mixtures. NMR methods are also useful in the investigation of metabolite molecular dynamics and mobility, as well as substance concentrations. This can be done by interpreting the spin relaxation times of NMR and determining the molecular diffusion coefficients.

Spin is a fundamental property of nature. It is a characteristic of protons, electrons and neutrons. Spin values are multiples of  $\frac{1}{2}$ , either positive (+) or negative (-). Particles whose spins have opposite signs pair up, eliminating the effects of spin. Each unpaired electron, proton and neutron possesses a spin of  $\frac{1}{2}$ . In NMR, these are the important nuclear spins. Similarly to electrons, nucleons form orbitals. As they also possess spin, their spin can also pair up when their orbitals are being filled and hence cancel out. Most elements in the periodic table have an isotope which possesses a non-zero nuclear spin. NMR is possible only when these isotopes exist in high enough natural abundance to allow them to be detected. A number of well-known nuclei used in NMR and their spin can be seen in Table 3.1 below:

**Table 3.1:** Examples of nuclei and their spin

Nuclei	Unpaired protons	Unpaired neutrons	Net Spin
$^1H, ^{19}F, ^{31}P$	1	0	$\frac{1}{2}$
$^{13}C$	0	1	$\frac{1}{2}$
$^{127}I$	2	1	$\frac{3}{2}$
$^{11}B, ^{23}Na, ^{35}Cl, ^{79}Br$	1	2	$\frac{3}{2}$
$^{17}O$	2	3	$\frac{5}{2}$
$^2H, ^{14}N$	1	1	1

Examples of nuclei with zero spin are  $^{12}C, ^{16}O, ^{32}S$ . The nuclear magnetic moment of a nucleus can have only specific values related to the nucleus spin. The value of the magnetic moment,  $\mu$ , of the nucleus is given by

$$\mu = \frac{\gamma h I}{2\pi} \quad (3.3.1)$$

where  $h$  is Planck's constant equal to  $h = 6.625 \times 10^{-34} \text{ Joule}\cdot\text{sec}$ ,  $I$  is the net spin of the nucleus and  $\gamma$  is the gyromagnetic ratio, which depends on the nature of each nucleus. The energy of a spin in a magnetic field will depend on the magnetic field, denoted

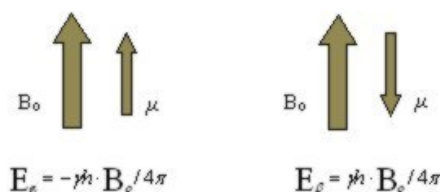
$B_o$  here. Applying this external magnetic field to the nucleus will cause its magnetic moment to align with the applied field in only  $2I + 1$  ways, either with or against the applied field. The energy of the spins is the dot product of the corresponding vectors:

$$E = -\mu B_o. \quad (3.3.2)$$

Thus, the energy difference of the two levels (Figure 3.14),  $\alpha$  and  $\beta$  is

$$\begin{aligned} \delta\epsilon &= E_\beta - E_\alpha \\ &= \frac{\gamma h B_o}{4\pi} - \left( -\frac{\gamma h B_o}{4\pi} \right) \\ &= \frac{\gamma h B_o}{2\pi}. \end{aligned} \quad (3.3.3)$$

From equation 3.3.3, it is obvious that the larger the magnetic field  $B_o$  is, the larger the



**Figure 3.14:** Spin energy levels

energy difference it will be e.g. pp 250, Table 1 in (Kealey and Haines, 2002). Nuclei are placed in one of the spin states. The number of nuclei in each spin state, that is, the population ratio between the two energy levels, depends on the energy difference of the two levels,  $\delta\epsilon$ , and it can be calculated by the Boltzmann distribution as

$$\frac{N_\alpha}{N_\beta} = e^{\frac{-\gamma B_o}{\kappa T}}, \quad (3.3.4)$$

where the  $N$  values are the number of nuclei in each one of the spin states,  $\gamma$  is the magnetogyric ratio,  $B_o$  is the external magnetic field strength,  $\kappa$  is the Boltzmann constant, and  $T$  is the temperature ( $K$ ).

### 3.3.3 History of NMR

NMR spectroscopy is a technique based on the fact that the nuclei of atoms have a physical property called the magnetic moment. In 1924, Pauli postulated that there are specific nuclei which possess a spin angular momentum. As a consequence, Gerlach and Stern confirmed through experimentation that nuclei have magnetic moments (Emsley and Feeney, 2007). Gorter was the first to perform an NMR experiment in solid state in 1936, though it was not successful. The first demonstration of how molecules in a constant magnetic field affect an oscillating electromagnetic field by absorbing resonance

was achieved by Rabi in 1939. NMR spectroscopy was first developed and used in experiments successfully independently by two groups, at Stanford University by Bloch, and Harvard University by Purcell, in 1946. Four years later, Hahn replaced the standard, until then, continuous wave excitation of polarised nuclei by pulse excitation. Chemical shifts<sup>14</sup> were discovered in 1951 by Arnold, who obtained the first high-resolution spectra. Technology limitations did not allow Arnold's idea of pulse spectroscopy to mature until 1960, when Anderson and Ernst were able to apply Fourier Transformations using computers. It was then possible to change time domain to frequency domain in one keystroke (Emsley and Feeney, 2007).

In the 1970s, NMR was used for the first time in medical applications. During the years 1970-73, Lauterbur showed that it was feasible to use NMR for imaging. He applied gradients to encode the spatial information into an NMR spectrum. In addition, Damadian discovered in 1971 that tissue contrast was available through variation of nuclear relaxation times (Emsley and Feeney, 2007). Both these advances proved fundamental in the use of NMR in medical applications. NMR spectroscopy is also used extensively, in many different fields such as in physics, chemistry, biochemistry, geology, agriculture and archaeology.

### 3.3.4 Description of a NMR Spectrometer

The main components of an NMR spectrometer are the *magnet*, the *frequency generator* (which creates the alternating current at Larmor frequency  $\omega_0$ ), the *detector*, responsible for subtracting the base to the output frequency, and the *recorder*, which includes the computer and the console parts (Figure<sup>15</sup> 3.15).

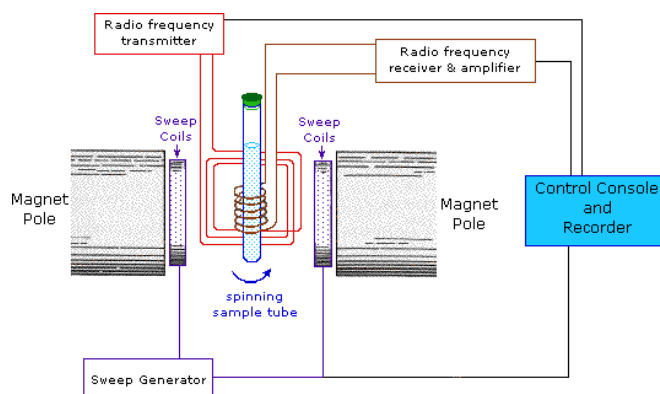


Figure 3.15: The main components of a NMR spectrometer

<sup>14</sup>More specifically  $^1H$  chemical shifts

<sup>15</sup>Source:<http://www2.chemistry.msu.edu/faculty/reusch/VirtTxtJml/Spectrpy/nmr/Images/spctmtr.gif>

### 3.3.4.1 The Magnet

This is usually a superconducting magnet and one of the most expensive components of the NMR system. Its main use is the production of the  $B_o$  field which is necessary for NMR experiments. The magnet includes the shim coils, the probe, the sample and the RF coils. A superconducting magnet has an electromagnet made of superconducting wire. By inserting the wire into liquid helium, its temperature drops to near absolute zero ( $0K$ ), which reduces its resistance to almost zero. Current will continue to flow in the coil for as long as the coil is kept at liquid helium temperatures. This wire is wrapped in a multi-turn solenoid or coil. The coil and liquid helium are kept in a dewar. This is surrounded by a liquid nitrogen dewar, acting as a thermal buffer between the room temperature air ( $293K$ ) and the liquid helium. A graphical description of a superconducting magnet with the concentric dewars can be seen in Figure 3.16.

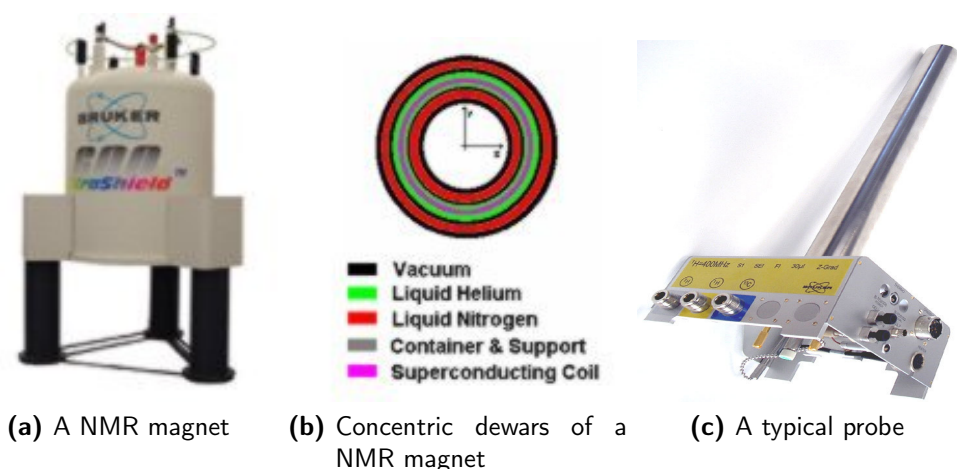


Figure 3.16: Example of a NMR magnet

The *shim coils* are exactly within the bore of the magnet, and their purpose is to adjust any minor spatial inhomogeneities that might exist in the  $B_o$  magnetic field. Reasons for these inhomogeneities could be the magnet design, materials in the probe, any variations in the thickness of the sample tube and the sample's permeability. Shim coils are designed to create small magnetic fields to oppose and eliminate any inhomogeneity in the  $B_o$  magnetic field. As there are inhomogeneities in a great variety of functional forms (linear, parabolic, cubic, etc.), shim coils must be able to create a variety of opposing magnetic fields. Usually the computer is responsible for controlling the shim coils.

The *probe* is one of the most important parts of the magnet (and of the NMR spectrometer). It is responsible for delivering RF radiation to the sample and receiving the signals coming from the sample. In Figure 3.16, a typical sample probe head and part of the probe tube can be seen. The probe is inside the shim coils and contains the *RF coils*, the *sample*, the *sample spinner* and the *temperature controlling circuitry*. The



*sample spinner* is used for the rotation of the NMR sample tube around its axis. In this way, each spin at a given position along the  $Z$  axis and radius from the  $Z$  axis experiences the average magnetic field in the circle defined by this  $Z$  and radius. As a consequence, there will be a narrower spectral line width. The *RF coils* are responsible for the creation of the  $B_1$  magnetic field which rotates the net magnetisation in a pulse sequence. In addition, they detect the transverse magnetisation as it unfolds in the  $XY$  plane. They usually act as the transmitter of the  $B_1$  field and receiver of RF energy from the *sample*. There can be one or more RF coils in a probe.

### 3.3.4.2 The Detector

This is a device used for the separation of the  $M_{x'}$  and  $M_{y'}$  signals from the signal received from the RF coils. The main component of the detector is a device called a doubly balanced mixer. This mixer has two inputs and one output. For example, assuming two input signals,  $\text{Cos}(A)$  and  $\text{Cos}(B)$ , the output will be the product of these two signals, that is  $\frac{1}{2}\text{Cos}(A+B)$  and  $\frac{1}{2}\text{Cos}(A-B)$ . The detector usually contains two doubly balanced mixers, two filters, two amplifiers and a  $90^\circ$  phase shifter. The detector has two inputs and two outputs. These inputs are the frequencies and the components of the transverse magnetisation are the outputs of the device.

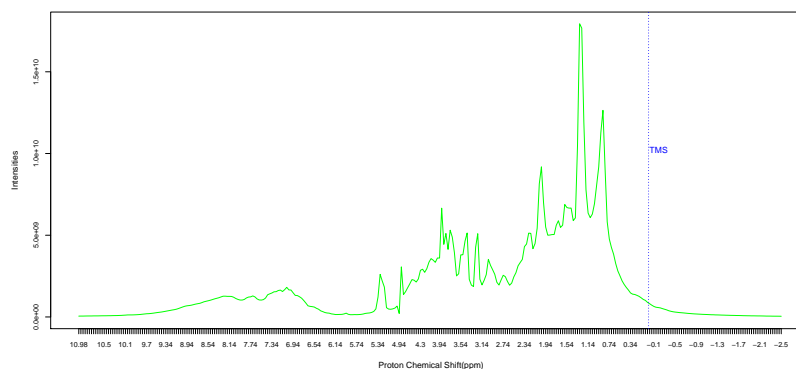
### 3.3.5 Description of an NMR Spectrum

An NMR spectrum is a graphical depiction of a living organism's biochemical (biomolecular) profile. It consists of an absorption line of the resonances signals of the chemical structures in the organism's profile. The chemical structures are positioned in this line, according to their nuclei chemical shifts. That is, according to the exact position of their resonance frequency (Kealey and Haines, 2002). To measure the chemical shift of a compound, a dimensionless parameter  $\delta$  is used. The compound's chemical shift is measured with respect to a reference compound which is usually *tetramethylsilane (TMS)* because it is inert, volatile, non-toxic and cheap, having only one low frequency resonance signal being at lower frequency than those of most organic compounds (Williams and Fleming, 1995). The chemical shift scale  $\delta$  is then defined by the following formula (Kealey and Haines, 2002):

$$\delta = \frac{\nu_{\text{compound}} - \nu_{\text{TMS}}}{\nu_{\text{spectrometer}}} \times 10^6$$

where  $\nu_{\text{compound}}$  is the resonance frequency of the required compound,  $\nu_{\text{TMS}}$  is the resonance frequency of the reference compound and  $\nu_{\text{spectrometer}}$  is the operating frequency of the spectrometer in use. The ratio above is multiplied by  $10^6$  to obtain easier to handle numerical values. Therefore,  $\delta$  is expressed in terms of fractions of the applied magnetic field in parts per million (p.p.m.) (Williams and Fleming, 1995). The chemical shift of

the reference compound is assigned a value of zero, thus  $\delta$  values for all the compounds in the spectrum are presented as a scale which increases from right to left. The greater the shielding of the nucleus of the compound, the smaller the value of  $\delta$  is and the further to the right the resonance signal appears and conversely (Kealey and Haines, 2002). It should be noted that  $\delta$  is independent of the operating frequency of the spectrometer, therefore chemical shifts in ppm from spectra recorded in spectrometer with different operating frequencies can be compared. In a proton NMR spectrum, usual  $\delta$  values are in the range 0 – 11. An example of a proton NMR spectrum of blood serum from the epileptic data can be seen in Figure 3.17. In a proton NMR spectrum, specific



**Figure 3.17:** Proton NMR spectrum of patient 43 from the epilepsy data set. All metabolites' chemical shifts in the spectrum have been measured with respect to the reference compound *tetramethylsilane* (TMS), whose chemical shift has been assigned a value of zero.

molecular groups can be identified at specific chemical shift values. More specifically, the aliphatic group is between 0 – 2 ppm, the Acetylenic between 2 – 4 ppm, the Olefinic in the range 4 – 6 ppm, the Aromatic group in the range 6 – 8 ppm and the Aldehydic between 8 – 10 ppm.

### 3.3.6 Applications of NMR to Metabonomics Studies

NMR metabonomics analyses are concerned with the detailed investigation of biomolecular reactions to the metabolic profiles of living organisms. This is achieved by analyzing biofluids and tissues such as blood plasma and serum, urine, cerebrospinal fluid (CSF), seminal fluids, bile, cardiac/brain/liver tissue and many others. NMR spectroscopy of metabonomics applications can be classified in general, with respect to their type of research, into the following areas: clinical applications (disease/disorder identification and classification), xenobiotic toxicity (application of various toxins to living organisms) and physiological variation (influence of physiological factors to the biochemical composition of living organisms) (Lindon et al., 2001; Antti et al., 2002).

There are a large number of independent clinical studies in an extensive array of diseases/disorders. Such studies include among many others the identification of dia-

gnostic biomarkers in the metabolic profiles of patients with *type I diabetes* (Makinen et al., 2008), the identification of biomarkers for the severity of the disease in patients with *rheumatoid arthritis (RA)* (Lauridsen et al., 2010), the investigation of the role of dystrophin to the metabolic profiles of brain, cardiac and muscle tissue of mice with *Duchenne muscular dystrophy (DMD)* and the characterization, detection and classification of cancer e.g. detection and separation of patients with epithelial ovarian cancer and benign ovarian cysts (Odunsi et al., 2005).

Xenobiotic toxicity is an important area of NMR application. This area involves the study of the effects of various chemical substances in the metabolic profiles of living organisms. Such studies are the *hepatotoxicity* and *nephrotoxicity* induced to rats treated with *copper nanoparticles* in different doses (Lei et al., 2008), the *hepatotoxicity* induced to rats with *allyl formate* (Yap et al., 2006) and the mercury toxicity from nephrotoxic lesions induced in Fischer 344 rats with  $HgCl_2$  and *2-bromoethanamine (BEA)* (Holmes et al., 1992).

Drug toxicity can also be assessed successfully using NMR metabonomics. Sussulini et al. (2009) studied the effects of different drug treatments to the metabolic profiles of patients with bipolar disorder and were able to distinguish patients treated with lithium from those treated with other medications.

Another area of NMR metabonomics applications involves the evaluation of the influence of various physiological factors to the metabolic profiles of living organisms. Such factors can be *intrinsic* and *extrinsic*. Intrinsic factors include hormonal effects (Bollard et al., 2005b), species differences (Bollard et al., 2005a), age-related differences (Bollard et al., 2005b), strain differences (Holmes et al., 2000), genetically modified models (Griffin et al., 2001), gender differences (Bollard et al., 2005b) and general inter-animal variation (Bollard et al., 2001; Zuppi et al., 1997). Extrinsic factors include diurnal effects (Bollard et al., 2001; Gavaghan et al., 2002), diet and fasting (Gavaghan et al., 2001; Solanky et al., 2005; Yde et al., 2010), water deprivation (Bollard et al., 2005b), temperature effects (Bollard et al., 2005b) and sleep deprivation/stress/acclimatization (Bollard et al., 2005b; Zuppi et al., 1997).

## 3.4 Comparison of Metabonomics Techniques

The main advantages and disadvantages of MS and NMR are given below.

### MS - Advantages/Disadvantages

The main advantages of MS are (Zhu et al., 2010; Kealey and Haines, 2002):

- High sensitivity. That is, it provides higher resolutions than NMR.
- Offers rapid detection of metabolites.
- Provides selective qualification and quantification of metabolites.

- It can simultaneously identify and measure a variety of metabolites.

However, in using MS techniques in "omics" studies, the researcher can meet the following problems (Zhu et al., 2010; Kealey and Haines, 2002):

- MS is a destructive analytical technique. That means, after an MS analysis the samples cannot be reused for other analyses.
- Before applying MS it is necessary to apply a number of different separation techniques depending on the classes of the substances to be analysed.
- The detection limits are lower if the substance to be analysed, can be ionised.
- MS methods require conformation from standard compounds, which is often not available especially for unknown compounds.

### NMR - Advantages/Disadvantages

The main NMR advantages are (Zhu et al., 2010; Lindon et al., 1999; Williams and Fleming, 1995):

- It is a non-destructive technique. After an NMR analysis, the samples can be reused for other analyses.
- In "omics" studies involving complex biomixtures, measurements can often be made with minimal sample preparation.
- The objective compounds can be qualified.
- NMR can provide detailed information on molecular structure for pure compounds and complex mixtures.
- It can provide information on absolute or relative concentrations.
- It can be conducted in vivo on whole live organisms, which is useful when metabolic profiling for studies of diseases is required.
- It is particularly useful for distinguishing isomers, for obtaining molecular information and for studies of molecular dynamics and compartmentation.

There are some disadvantages when applying NMR to "omics" studies (Zhu et al., 2010; Lindon et al., 1999):

- NMR suffers from an intrinsic low sensitivity for low concentrations of metabolites.
- Chemical noise, as a result of the overlapping of signals from compounds low in abundance, can significantly reduce the amount of recoverable spectral information especially in NMR analysis of biofluids.

The last chapter of Part I of the thesis, covers the most popular pre-processing and pre-treatment methods for the enhancement of the quality and accuracy of the metabonomics data, and the preparation of the data in a suitable form for further statistical analyses, in NMR spectra.

# Chapter 4

## Pre-processing and Pre-treatment of the Data

### 4.1 Introduction

A very important part of any chemometrics data analysis, called *pre-processing* and *pre-treatment*, is the application of certain operations to the data in order to either remove or at least reduce to an acceptable point, the amount of random or systematic variation for which the main modelling tool is not responsible. Pre-processing of a data set is the general term for those methods used to convert the raw instrumental data to clean data for pre-treatment and further processing (Goodacre et al., 2007). These methods include *Binning*, *Deconvolution*, *Peak Detection*, *Alignment* and *Baseline Corrections* among others. Pre-treatment involves the transformation of the clean (pre-processed) data to prepare it for data processing. Metabonomics data are mostly presented in tabular form, with each row of such a table relating to a specific sample and each column to a single measurement (or variable). Pre-treatment methods include scaling operations to the rows (row-scaling), the columns (column-scaling) and to individual elements of a data set (transformations) and the most common such operations are the *mean-centring*, *vector normalisation*, *autoscaling* and *logarithmic transformation*. Pre-processing and pre-treatment of the data most of the time has either positive or negative effects on the results of the analysis. This chapter describes the most important and commonly used pre-processing and pre-treatment methods in metabonomics data. The definition of the *sensitivity* of a measurement in NMR is given in Section 4.2. Pre-processing methods are described in Section 4.3 and pre-treatment methods can be found in Section 4.4.

## 4.2 Sensitivity of a Measurement in NMR

According to [Ross et al. \(2007\)](#), the sensitivity of a measurement of an NMR experiment, defined as the signal-to-noise ratio ( $\frac{S}{N}$ ) of a single repetition of the experiment, is given by:

$$\text{Sensitivity}_{scan} \sim \gamma_X^3 \cdot N \cdot B_o^2 \cdot S_D$$

where  $\gamma_X$  is the gyromagnetic ratio,  $N$  the number of spins in the sample,  $B_o$  the magnetic field strength and  $S_D$  the sum of the sensitivity of the detector and the noise created by the sample. The higher the sensitivity of the measurement is, the better the resolution of the spectra generated and the more detailed the information about metabolites will be. To increase the sensitivity of measurements, either the sensitivity  $S_D$  needs to be increased or a higher static magnetic field (higher  $B_o$ ) is required. The former can be increased either by using cryogenic probes or by reducing the size of the detection coil, as the shorter the wires are the less noise they produce. The latter is a matter of the NMR magnet specification in use, thus it is not adjustable by the researcher.

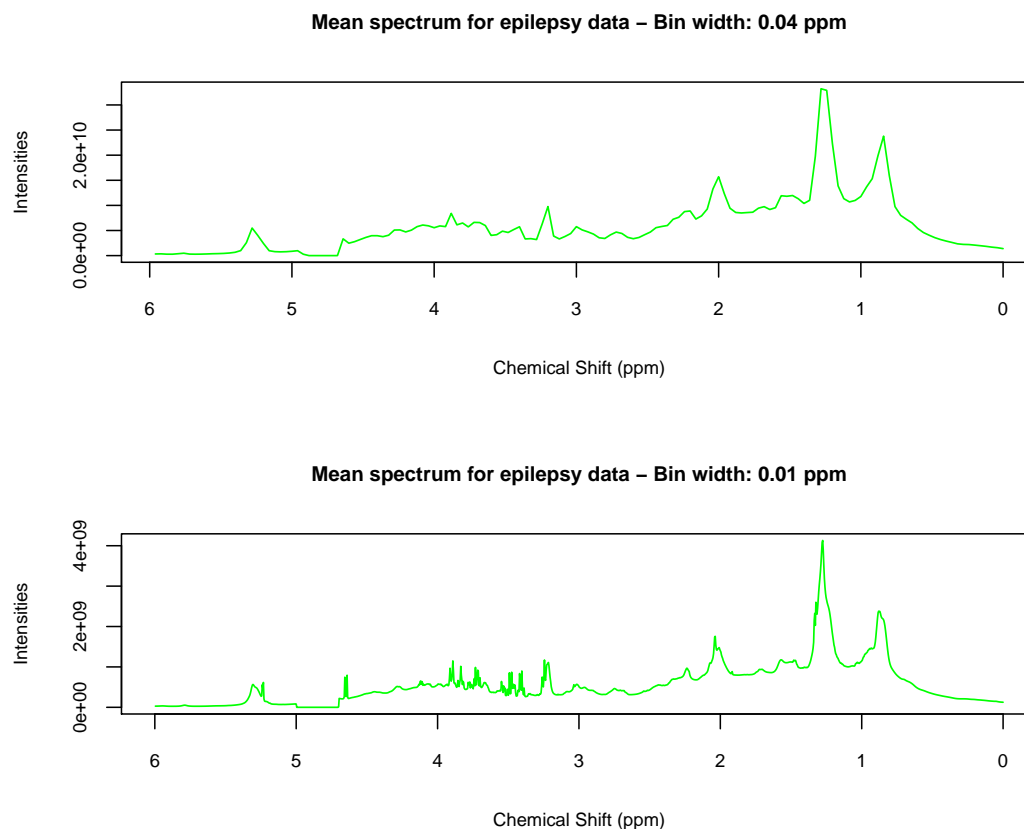
## 4.3 Pre-processing Methods

After generating the signals, it is often necessary to apply specific techniques to *clean* the data. More specifically, problems that can appear to the generated signals include among others overlapping peaks, misalignments of metabolites in the spectra, signal phasing, baseline drifts, as well as very large numbers of metabolites in the data. These methods, also called *signal processing* methods, include various operations which can be applied to the spectra. The most commonly used are described in the following sections.

### 4.3.1 Binning (Bucketing)

Binning is a pre-processing method which in general, is used to integrate the variable signals into specified segments of a MS or NMR spectrum. Its use is important for spectral resolution tuning and offers a proper representation of the data for further processing ([Craig et al., 2006](#); [Boccard et al., 2010](#)). In MS, the ion intensities are integrated with respect to  $\frac{m}{z}$  and retention time intervals. Thus, the data comprises a  $2 - D$  array such that one of the axes represents the retention time values and the other the  $\frac{m}{z}$  values. Each value in this array is a measured intensity for a given  $\frac{m}{z}$  at a given retention time. Although there is an unavoidable loss in resolution, the binned data is represented in a way that means it is easier to handle and process further ([Boccard et al., 2010](#)). In NMR-based data, binning of spectra involves the integration of peak values within specified spectral ranges ([Craig et al., 2006](#)). Two binning procedures can

be used in NMR spectra. These are the *equidistant* and the *non-equidistant* binning (Ross et al., 2007). In the former, after creating a peak frequency and intensity listing, the spectrum is divided into regions (called "bins or "buckets") of defined width e.g. 0.01, 0.04, 0.05 ppm, and the peak heights are summed up in each region to obtain a series of numerical descriptors equally spaced along the NMR frequency (chemical shift) axis (Spraul et al., 1994). For example, in the case of the bin width being 0.04 ppm, a bin at the chemical shift of 1.3 ppm would include the sum of the intensity values of all peaks with chemical shifts in the range 1.3 – 1.33 ppm, and these peaks would be represented in the spectrum with one peak (one point) at 1.3 ppm. The latter type of binning aims to prevent peaks being cut by the boundaries of bins. In this case, the borders of the bins are adjusted such that the bins cover only complete peaks including all possible locations of the peaks. Hence, the bin width depends on the width of the peak shape and on the shift width of the peak. An example of equidistant binning of the epilepsy data can be seen in Figure 4.1. The original raw epilepsy spectra contain many



**Figure 4.1:** Example of binning the epilepsy data. The spectra in both plots are the mean spectra of the 97 patients in the epilepsy data. The top plot contains 144 bins of width 0.04 ppm, whereas the bottom 1500 bins of width 0.01 ppm.

thousand metabolites. After the binning procedure is applied, the resulted spectra for bin width 0.04 ppm contain 144 variables (bins), while those for bin width 0.01

*ppm* contain 1500 variables (bins). However, only the full resolution spectra should be used for biomarker identification, whereas the binned data should be used for the development of classification models (Craig et al., 2006).

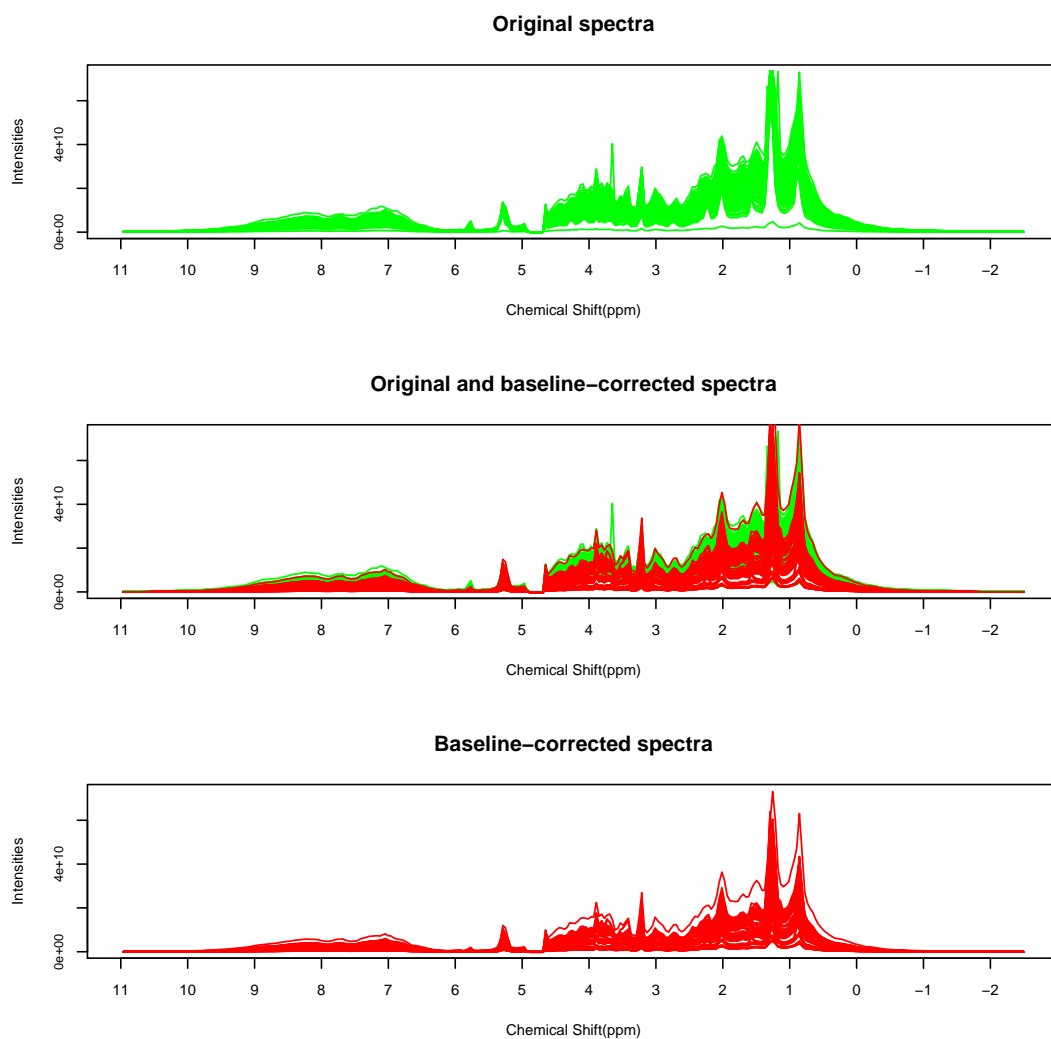
### 4.3.2 Baseline Correction

Often in NMR and MS experiments, the generated spectra may appear to show baseline inconsistencies. Baseline offsets from spectrum to spectrum can affect the results of the data analysis in many ways. They affect negatively the abundance (MS) and the intensity (NMR) values, hence causing problems in the accurate peak assignment and quantification (Xi and Rocke, 2008). For example, in a PCA model, baseline effects may cause the introduction of extra components in the model and as a consequence the results and interpretation of the analysis could be significantly altered from those taken from the actual (corrected) model (Gemperline, 2006). Especially when metabonomics data are concerned in a study, it is even more important to correct any baseline problems, as they usually contain many small but potentially statistically significant peaks which are sensitive to baseline offsets. This can cause a failure in detecting important metabolites or even in identifying potential biomarkers (Xi and Rocke, 2008). There are many different types of baseline effects varying from a simple offset to extremely complex shapes such as an upward or downward sloping line or even a broad curved shape. The ways to remedy these problems, depend on the type of baseline error in the spectra. In simple offset cases, knowing that a specific region in the spectra has signal values equal to zero, it is usually sufficient to subtract the average value of the signal in this region (chemical shift or  $\frac{m}{z}$ ) for each spectrum, from each metabolite in the respective regions. In more complex cases, it may be necessary to fit a polynomial function through all the valleys in the spectra. This polynomial line is then subtracted from the corresponding spectrum to correct the baseline differences (Gemperline, 2006). These methods are also called as *frequency domain correction* methods (Xi and Rocke, 2008). An example of the application of a baseline correction method (asymmetric least squares, (Eilers, 2004)) can be seen in Figure 4.2.

### 4.3.3 Deconvolution

A common problem in NMR and MS metabonomics studies is the appearance of overlapping peaks in the spectra. *Deconvolution* is a pre-processing technique which is used to overcome this difficulty (Goodacre et al., 2007). Due to the fact that fragments, adducts and molecule isotopes increase the difficulty in detecting peaks in the signals, it is necessary to improve the detection procedure. The process of applying deconvolution algorithms is similar to both analytical techniques. In MS, it is necessary to use the profile resolutions of both spectral and chromatography steps. By correlating the





**Figure 4.2:** Illustration of baseline correction of the epilepsy metabolite spectra. The top figure illustrates the original spectra without baseline-corrections, while the bottom figure illustrates the original spectra after baseline-corrections. In the middle figure, the baseline-corrected spectra are superimposed to the original spectra. The outlying low green line in the top figure is the spectrum with the lowest concentration in metabolites, whereas the outlying high red line in the middle and bottom figures, the spectrum with the highest concentration in metabolites.

sample profiles with the retention time, the aim is to regroup ions coming from the same metabolite. This procedure may not be so efficient when metabonomics analyses take place due to the increased complexity of the data. Metabonomics data can contain a large number of highly correlated signals at similar retention times which leads to overlapping peaks with close biological variations (Boccard et al., 2010). There are many algorithms which can be used to improve the spectra resolution such as the *Gold*, the *Richardson-Luey*, the *Fourier* and the *Van Cittert* algorithms.

### 4.3.4 Smoothing

In a given spectrum, apart from the true signal, there is always an amount of unwanted random noise. The type and amount of this noise depends on the experiment. Smoothing operations are necessary to increase the signal-to-noise ratio, that is to reduce the amount of that noise in the spectrum. The most common smoothing methods use the mean, the running mean, the running median, the running polynomial and the Fourier filter smoother. The mean smoother is suitable for the reduction of the number of variables, but it can cause problems, as it reduces the resolution, and so may eliminate important information. The running polynomial smoother is the most suitable of the three running smoothers, for noise reduction, although the running median smoother is better used when there are high-frequency spikes for removal. The Fourier filter smoother is suitable for general smoothing but there are specific requirements to be met for its use (Beebe et al., 1998).

## 4.4 Pre-treatment Methods

### 4.4.1 Introduction

Once pre-processing of the data has been completed, it is quite often necessary to apply also pre-treatment methods, to prepare the data for processing. As it is common in metabonomics data analyses, part of the observed variation is uninduced due to biological and technical (sampling, sample work-up and analytical measurement errors) variation. In addition, the data is more often than not heteroscedastic. Data pre-treatment methods are used to reduce as much as possible the effects of these problems. These methods depend both on the required biological information and on the processing method to be used for the statistical analysis of the data. Pre-treatment methods can be applied to the rows (row-scaling), to the columns (column-scaling) and to individual elements of a data set, called transformations (Brereton, 2009; Berg et al., 2006).

### 4.4.2 Scaling

Scaling methods are pre-treatment operations used to adjust the importance of the various elements in the data to the model-fitting procedure. The adjustment usually involves the weighting of the metabolites with a factor which can be estimated by using either a dispersion criterion or a size measure (Boccard et al., 2010).

#### Centring

In general, centring pre-treatment methods allow the researcher to focus on the differences and not the similarities in the data. They are concentrated in isolating and

removing the systematic variation in the data. Care is needed when data are heteroscedastic, as the effects from centring methods might not be sufficient. Usually centring methods are applied in combination to the other pre-treatment methods. They belong to the column-scaling methods (Goodacre et al., 2007). The following are the most commonly used column-scaling methods in metabonomics.

- **Weighted (General) Centring.** It aims to convert all metabolite concentrations to fluctuations around zero instead of around their mean. Therefore, it retains only the relevant variation (the variation between the samples) for the analysis. It is also called *reference subtraction*. This method is particularly useful in PLS-DA classification where it can take care of several classes with different numbers of samples in each class. The weighted mean for a data set with  $N_c$  classes can be estimated as

$$\bar{x} = \frac{\bar{x}_g + \frac{\sum_{h \neq g} \bar{x}_h}{N_c - 1}}{2}$$

where  $\bar{x}_g$  and  $\bar{x}_h$  are the mean vectors for groups  $g$  and  $h$  respectively (Brereton, 2009). For two classes,  $N_c = 2$ , the above formula becomes

$$\bar{x} = \frac{\bar{x}_1 + \bar{x}_2}{2}$$

where  $\bar{x}_1$  and  $\bar{x}_2$  are the mean vectors for groups 1 and 2 respectively, and  $\bar{x}$  a global mean (but not the overall mean, which may be biased in favour of one of the two classes, especially when the two classes have different sample sizes). *Weighted* centring can then be achieved by subtracting the weighted mean from each column of the data set as long as there are  $N_c$  classes in the column.

- **Mean Centring.** This is a centring method, in which each column of the data, is expressed in deviations from its mean. In this way, the mean of the columns is subtracted, translating the centre of gravity of the dataset to the origin. The formulae for *mean* centring is

$$\tilde{x}_{ij} = x_{ij} - \bar{x}_j$$

where  $\tilde{x}_{ij}$  represents the data after *mean* centring and  $\bar{x}_j$ , the overall mean of variable  $j$ , is estimated as

$$\bar{x}_j = \frac{1}{N_{samples}} \sum_{i=1}^{N_{samples}} x_{ij}$$

with  $N_{samples}$  the number of samples in the data set.

### Vector Normalisation

Quite often the data of the various samples cannot be directly compared to each other, being recorded on different scales (e.g. different injection volumes in chromatography). To remove or minimize the variability from sample to sample, normalization of the samples can be applied. This operation puts all the samples on the same scale, thus allowing for comparisons among the various samples. *Normalization* involves dividing each variable of a sample vector by a constant. There are many different constants that can be used, such as the 1-norm and the 2-norm of the vector (Beebe et al., 1998; Craig et al., 2006; Brereton, 2009). For example, the 1-norm vector normalisation is given by

$$\tilde{x}_{ij} = \frac{x_{ij}}{\sqrt{\left( \sum_{j=1}^{N_{metabolites}} x_{ij}^2 \right)}}$$

so that the sum of squares of the elements of vector  $x_i$  after the *normalization* is equal to one. The selection of the appropriate normalization constant depends on the type of systematic variation in the samples. Normalisation belongs to the row-scaling methods (Brereton, 2009). It is an important step, as its purpose is to remove any systematic variation retaining all the biological information in the data.

### Scaling Based on Data Dispersion

These scaling methods use a dispersion measure for scaling the data and more specifically the columns of a data set (Berg et al., 2006; Goodacre et al., 2007; Boccard et al., 2010). In all these methods, the mean is defined as

$$\bar{x}_i = \frac{1}{N_{vars}} \sum_{j=1}^{N_{vars}} x_{ij}$$

and the standard deviation as

$$s_i = \sqrt{\frac{\sum_{j=1}^{N_{vars}} (x_{ij} - \bar{x}_i)^2}{N_{vars} - 1}}.$$

- **Autoscaling.** This is a form of scaling performed by mean-centring each metabolite value and using afterwards the standard deviation as the scaling factor. The formula is given by

$$\tilde{x}_{ij} = \frac{x_{ij} - \bar{x}_i}{s_i}.$$

*Autoscaling* is also called unit variance scaling or standardization, as after the *autoscaling* procedure, all metabolites have standard deviation equal to one, allowing the metabolites to be compared using correlations instead of covariances. Its main advantage is that all metabolites become equally important, but it can

allow for increase in measurement errors. After the application of *autoscaling*, the data becomes dimensionless.

- **Range Scaling.** The scaling factor in the *range* scaling method is the range within each metabolite. In this case, the formula is

$$\tilde{x}_{ij} = \frac{x_{ij} - \bar{x}_i}{x_{i_{max}} - x_{i_{min}}}.$$

*Range* scaling allows the comparison of metabolites with respect to their biological response range. In this way, all metabolites are equally important and their scaling is related to the biology of the data. However, increase in measurement errors and sensitivity to outliers may be noticed when applying this scaling method. As in the case of *autoscaling*, the data becomes dimensionless.

- **Pareto Scaling.** Here the square root of the standard deviation is used as the scaling factor. It aims to reduce the influence of large values without losing significant information concerning the structure of the data. It is estimated by

$$\tilde{x}_{ij} = \frac{x_{ij} - \bar{x}_i}{\sqrt{s_i}}.$$

*Pareto* scaled data is closer to the original than autoscaled data, but it depends very much on the large values in the data set.

- **Variable Stability (Vast) Scaling.** This is an extension of *autoscaling*. It aims to give more importance to those metabolites which appear to have small variance. To achieve that, the method uses the coefficient of variation (CV) statistic as scaling factor. The formula is given by

$$\tilde{x}_{ij} = \frac{(x_{ij} - \bar{x}_i) \cdot \frac{\bar{x}_i}{s_i}}{s_i}$$

where  $\frac{\bar{x}_i}{s_i}$  is the inverse of the coefficient of variation. This method is not useful when large induced variation exists and there is no group structure in the data.

All these scaling methods belong to the column-scaling methods, as the scaling is applied to the columns of the data set.

### Scaling Based on Average Value

These methods use a size measure instead of a spread measure. *Level* scaling is one such method. It converts the changes in metabolites concentrations into changes relative to the average concentration of the metabolite by using the mean concentration as the scaling factor. The resulting values are changes in percentages compared to the mean concentration. The formula for *level* scaling is given by

$$\tilde{x}_{ij} = \frac{x_{ij} - \bar{x}_i}{\bar{x}_i}.$$

This method is suitable for the identification of biomarkers. It is though prone to increase in the measurement errors. *Level* scaling, likewise the scaling methods based on data dispersion, belongs to the column-scaling methods.

An illustration of the effect of applying row and column-scaling on the PCA scores and loadings of the epilepsy data, can be seen in Figures 4.3 and 4.4 respectively. The scores plots in Figure 4.3 indicate that there are no differences among the scores of the six epilepsy data sets. With regards to the loadings plots in Figure 4.4, the loadings on PC1 are pretty well constant for the *autoscaled*, *range*, *Vast* and *level* scaled data. The loadings on PC2 for these four data sets have similar shape, although in the *Vast* scaling case, the loadings are mainly negative. In the other two data sets, the *true* and *Pareto*, the loadings on both PCs have similar shapes. In general, there is more variation in the loadings of PC2, and the shapes of the scores and loadings for the *true* and *Pareto* have the highest similarity among all plotted data sets.

### 4.4.3 Transformations

Metabonomics data can often be skewed and in general suffer from heteroscedasticity. In addition, the interactions between the various metabolites are not necessarily additive but can also be multiplicative (Boccard et al., 2010). As most of the statistical multivariate methods used for the analysis of metabonomics data often are more effective when the data is symmetric and many statistical significance tests more often than not assume that the distribution of the data is approximately normal, it is necessary to convert the data such that it approximates as closely as possible normality (Breton, 2009). Transformations of the elements of metabonomics data sets can help towards this aim. Three common transformation approaches are the *logarithmic*, the *power* and the *Box-Cox* transformation.

#### Logarithmic Transformation

It is achieved by replacing  $x_{ij}$  by  $\log(x_{ij})$ . The main advantages of applying this transformation are that it often minimises the problem with heteroscedastic data, converts multiplicative models to additive and reduces the influence of large data values such as outliers and occasional high peaks. However, its main weakness is its difficulty in handling zero or very close to zero values (especially when these values are very close to the limit of detection). If the values are below the limit of detection then they are considered as zero and therefore their logarithms are not defined (Breton, 2009; Berg et al., 2006).

#### Power Transformation

This is performed by replacing  $x_{ij}$  with  $x_{ij}^{(\frac{1}{n})}$  where for  $n = 2$  this is the square-transformation (square root) and so on (Breton, 2009). Advantages of using power

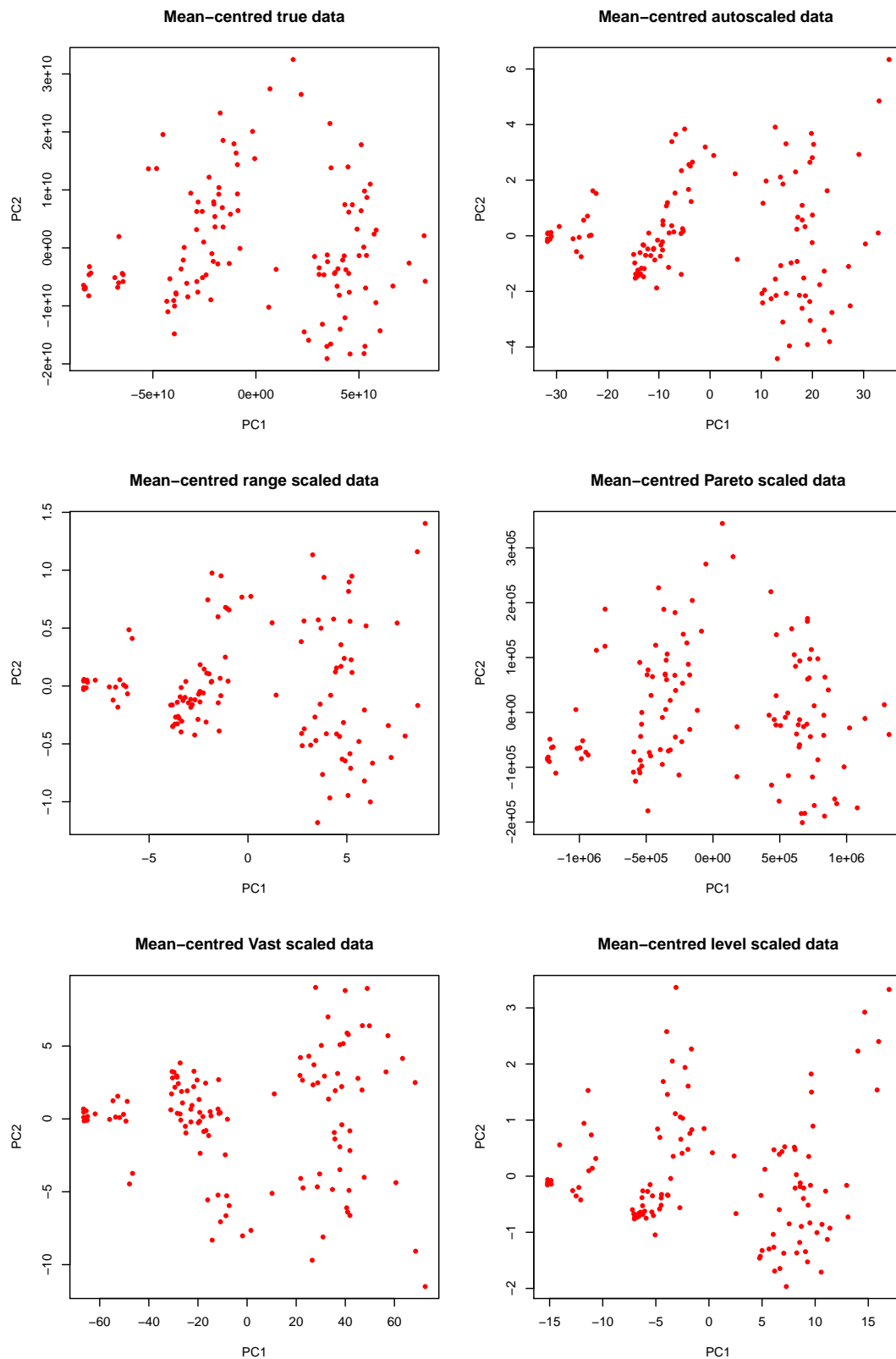


Figure 4.3: PC1 vs PC2 scores plots for the scaled epilepsy data sets.

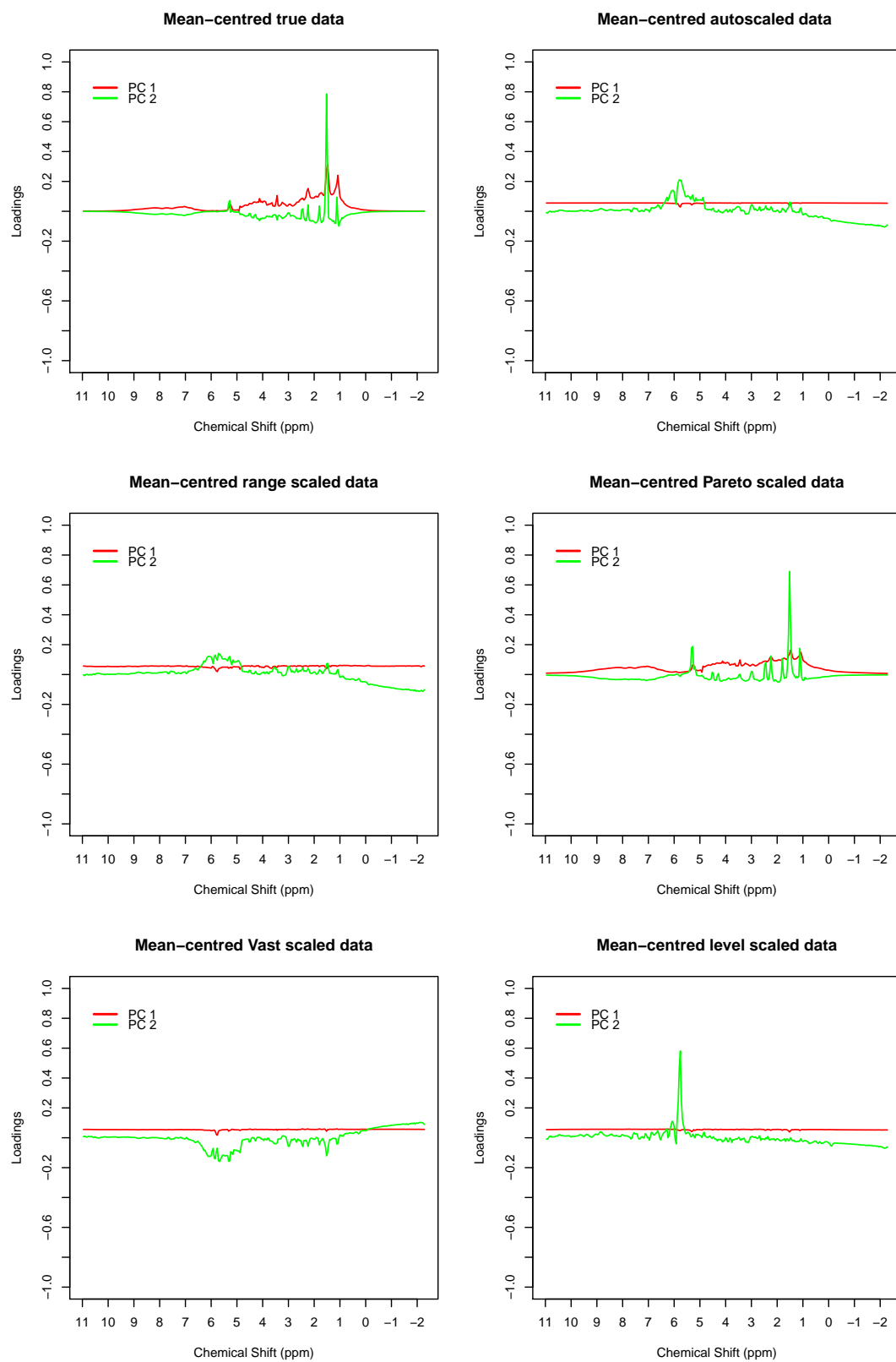


Figure 4.4: PC1 vs PC2 loadings plots for the scaled epilepsy data sets.



transformations are:

- It reduces the influence of large values such as outliers and occasional high peaks.
- Unlike the logarithmic transformation, it can cope with zero values, eliminating the need to replace values below the limit of detection.
- Any uncertainties in small values do not affect the data analyses as much as in the case of the logarithmic transformation. The smaller a value relative to other values is, the smaller its influence on the  $n^{\text{th}}$  root transformed data will be.

The drawbacks of this transformation can be summarised as:

- All values must be positive.
- If the distribution of the data is approximately log-normal, then power transformation cannot convert the distribution of values to a symmetric one.
- There are many options for the value of  $n$ . Trial and error is needed to identify the most appropriate choice for the root. Especially in multivariate data such as in metabonomics, where each metabolite may have a different distribution, it can be quite difficult to decide on the  $n$ .

### Box-Cox Transformation

In this case, the transformation is given by

$$x_{ij} \text{ transformed as } \begin{cases} \frac{(x_{ij}^\lambda - 1)}{\lambda} & \text{if } \lambda \neq 0 \\ \log(x_{ij}) & \text{if } \lambda = 0 \end{cases}$$

where  $\lambda$  is a real number (usually a non-integer, e.g. 0.3). The aim of this transformation is to convert the data into a normal distribution. This transformation is not as popular as the previously mentioned methods, since it may become very difficult to interpret the results of analyses of Box-Cox transformed data when the metabolites in the data set have each different distribution. This is especially true when further pre-treatment methods are applied to the data as the results can be very complicated and unpredictable. However, if the metabolites have similar distributions and there is no need for further pre-treatment of the data, then the Box-Cox transformation can be very efficient (Brereton, 2009).

An illustration of the effect of the various transformation methods on the PCA scores and loadings of the epilepsy data, can be seen in Figures 4.5 and 4.6 respectively. The scores plots in Figure 4.5 indicate that the scores of the six epilepsy data sets are quite similar in shape, with the scores of the *true*, *power* (for both  $n$  values) and *Box-Cox* (for  $\lambda = 0.8$ ) transformed data sets having the highest similarity. Concerning the loadings plots in Figure 4.6, there is a similar pattern to that of the scores, as the loadings on both PCs in the *true*, both *power* and the *Box-Cox* (for  $\lambda = 0.8$ ) transformed data sets are similar in shape. The loadings on PC1 are pretty well constant for the *log*

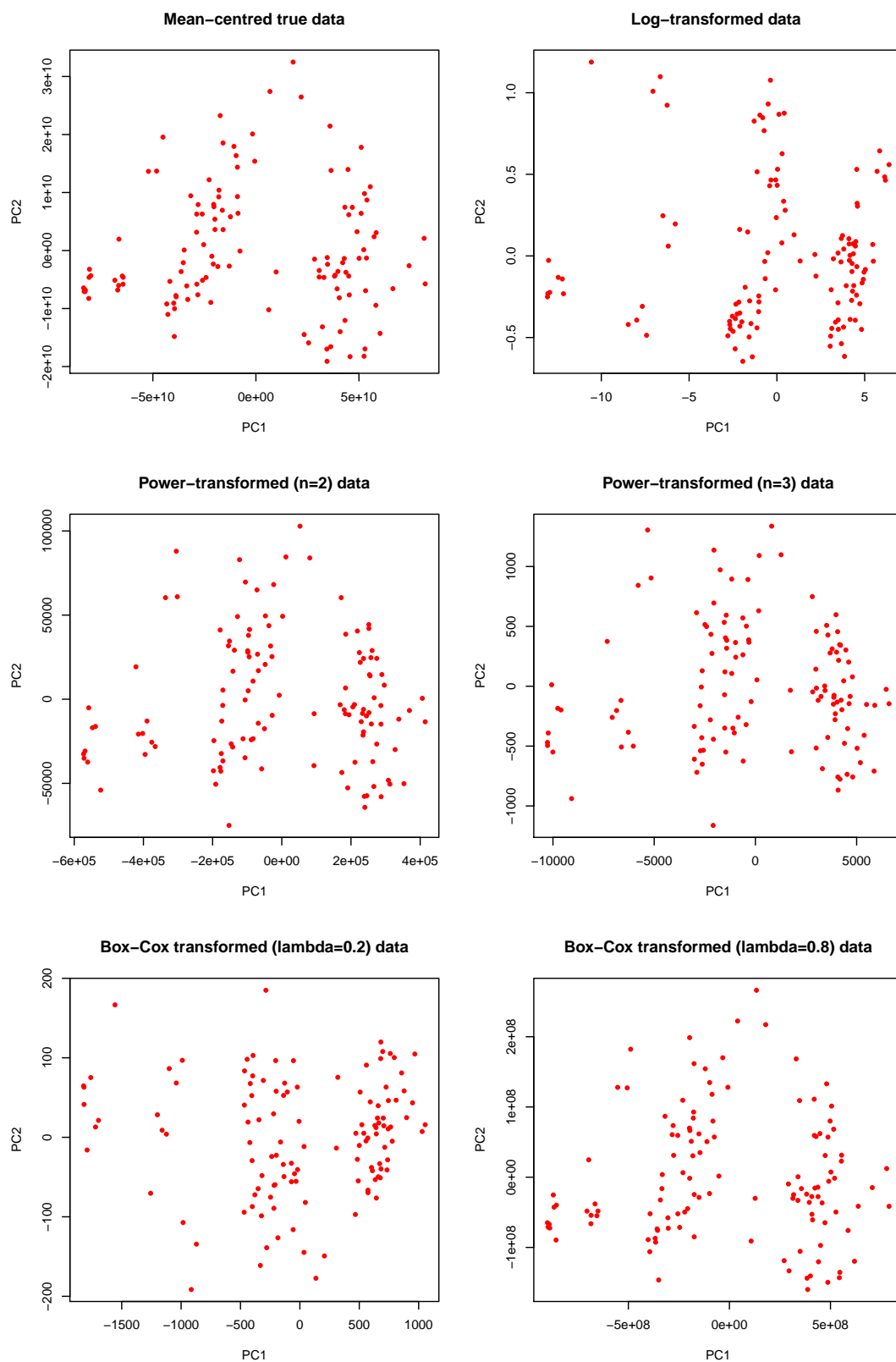


Figure 4.5: PC1 vs PC2 scores plots for the transformed epilepsy data sets.

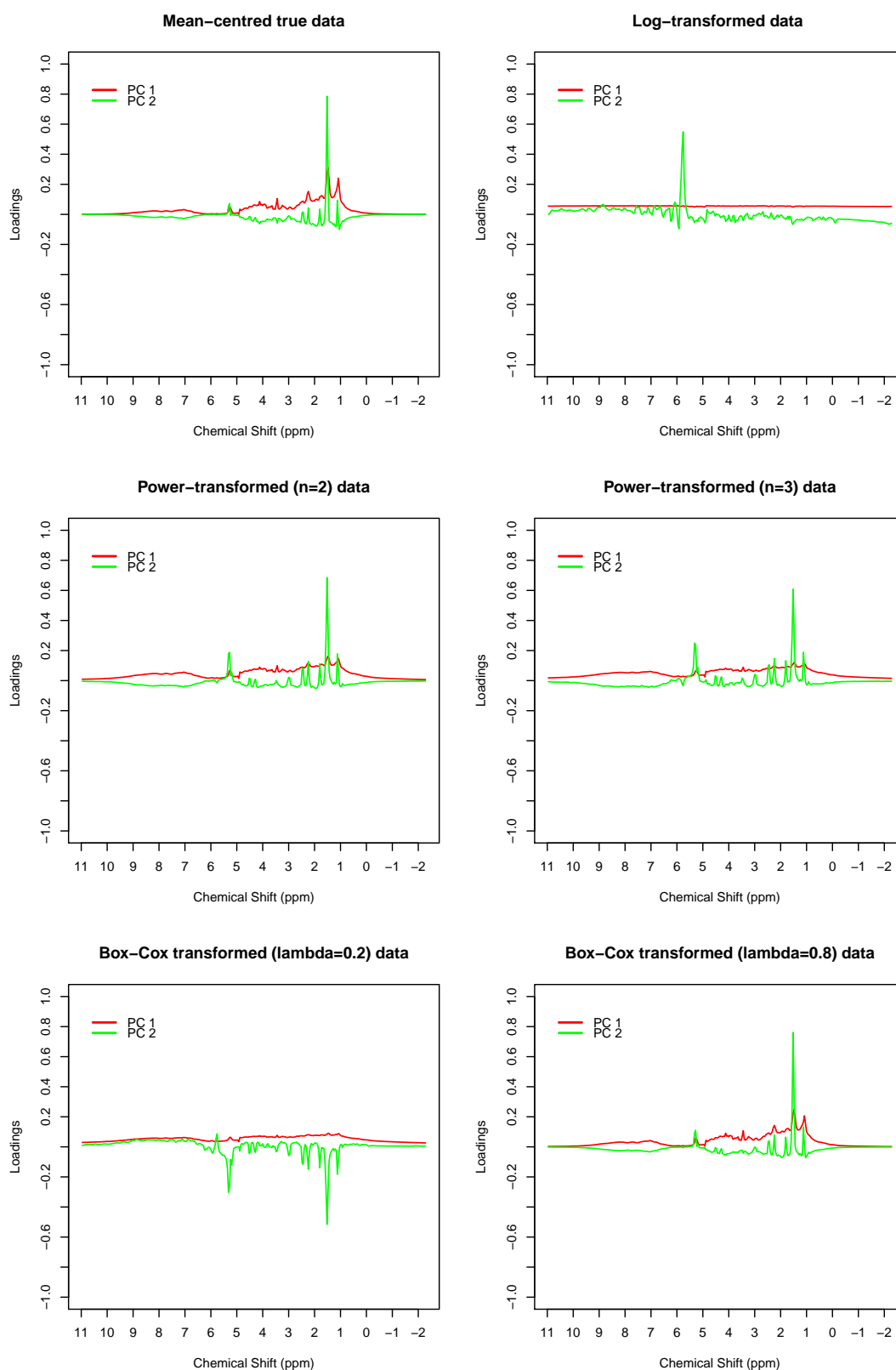


Figure 4.6: PC1 vs PC2 loadings plots for the transformed epilepsy data sets.

transformed data, whereas the loadings on PC2 for the *Box-Cox* (for  $\lambda = 0.2$ ) data are negative. In general, there is more variation in the loadings of PC2, and the shapes of the scores and loadings for the *true*, both *power* and the *Box-Cox* (for  $\lambda = 0.8$ ) have the highest similarity among all plotted data sets.

The normal order of performing the above mentioned pre-treatment methods in a data set is usually, first to transform the individual elements of the data set, then apply row-scaling and finally to scale the columns (Breton, 2009).

# Summary

In the introductory part of this project, the main aspects of generating and processing information about the metabolome have been discussed. The importance of the application of the *metabonomics* technology to facilitate the diagnosis of diseases as well as to evaluate the effects of drug treatment, was stated. Moreover, the main functional genomic levels (*transcriptome*, *proteome* and *metabolome*) and the technologies to study the functional networks and pathways of these levels were discussed briefly, with *metabonomics* and the analysis of metabolic networks and pathways being described in more detail. *Toxicogenomics*, which among others, is involved to the study of the way the genomes respond to drug treatment was also mentioned. The analysis of metabonomics data is achieved with multivariate statistical techniques, therefore the application of such techniques to metabonomics data, known as *chemometrics*, was also briefly mentioned in this part.

As the metabonomics data that will be used in this project for the purposes of the research scope are patients with epilepsy, a few sections were dedicated to describing this disorder. The definitions of *epileptic seizure* and *epilepsy*, as well as of *epilepsy syndrome* and *epilepsy disease* were given. Lists of the most common types of epileptic seizures, epilepsies and epileptic syndromes, as well as some important facts about the prevalence and cost of epilepsy worldwide and elsewhere, were stated. Afterwards, the problem to be researched was described in detail and the data set together with the main characteristics of the subjects, to be used in the statistical analyses of the problem were given.

To generate a metabonomics data set from the samples taken from the patients, an analytical technique must be used. Two such techniques, which are almost exclusively used to generate metabolic profiles, were described in detail, *Mass Spectrometry* (MS) and *Nuclear Magnetic Resonance Spectroscopy* (NMR). Information included the theoretical background of these techniques, historical milestones, a description of the main components of the appliances used to perform these techniques, the main steps occurring during the generation of the data, as well as a description of the output (spectra) generated by these two techniques after applying these procedures to the samples. For comparative purposes, the main advantages and disadvantages of using these analytical techniques are given.

Before any chemometrics analysis takes part, it is most of the times necessary to

process the generated metabonomics data to remove or reduce to acceptable levels the amount of systematic variation in the data. That is, to make the data more suitable for the statistical analyses to follow. There are two stages in the preparation of the data, the *pre-processing* and the *pre-treatment*.

*Pre-processing* is concerned with the cleaning of the generated signals, from problems such as overlapping peaks, baseline drifts, signal phasing and existence of an extremely large number of metabolites in the data. A range of methods to overcome such problems were briefly described with emphasis to those methods more suitable for NMR signals, as the spectra used in this project were generated by proton NMR spectroscopy. Examples of how these methods affect the appearance of the spectra were given for *binning* and *baseline correction*. However, the available spectra had been signal-processed by Dr. John Parkinson, therefore there was no need to apply any of the previously mentioned techniques.

*Pre-treatment* occurs in the second stage of data processing. *Pre-treatment* methods are always applied after *pre-processing*. Their purpose is to remove or reduce as much as possible any uninduced variation (due to sampling, sample work-up and analytical measurement errors) and if any, the heteroscedasticity of the data. Description of the most popular such methods were given with respect to the three ways that the methods can be applied to the data. *Row-scaling* scales the rows, *column-scaling* the columns, and *transformations* the elements of the data matrix. Scaling methods (both row and column) were classified to *centring*, scaling based on *data dispersion* and scaling based on *average values*. Three different types of transforming the elements of a data matrix were given, namely the *log*, the *power* and the *Box-Cox* transformation. Advantages and disadvantages of applying the above mentioned pre-treatment methods are given, as well as graphical representations (PC scores and loadings) of the effect of applying these to the first two PCs of the metabonomics epilepsy data. Using these results, and considering the type of data to be analysed in the project, the pre-treatment methods that were applied to the epilepsy data (resulting in the data set that was used for the exploratory analyses and clustering in Chapters 5-7), were *mean centring* for scaling the columns and *scaling to a constant total* the rows of the data matrix. More specifically, the elements of each column in the data matrix were transformed by subtracting the column mean from each element. Row scaling was achieved by dividing each column by the sum of all variables in each sample, resulting in all columns having sum equal to one, effectively making the columns more comparable to each other in the various analyses in Chapters 5-7. No element transformation was chosen to be used, since the results indicated that there was no significant (if any) improvement to the data by using any of the three previously mentioned transformation techniques.

In the next few Chapters (5-7), the research will be focussed to the application of the most commonly used and important unsupervised techniques in the metabonomics data

described in Chapter 2, with the processing (scaling and centring methods) mentioned above. These include both linear and nonlinear dimension reduction and visualisation methods such as *PCA*, *MDS/NLM* and *SOM*. In addition, unsupervised clustering techniques such as *HCA*, *Fuzzy* and *k-means* will be reviewed and applied to the selected metabonomics data. The data set was generated by the NMR spectroscopy analytical technique and it was pre-processed by Dr. Parkinson, as mentioned previously.

## Part II

# Pattern Recognition - Unsupervised Techniques



# Introduction

Due to the nature of the information contained in biological data sets (such as metabonomics data), high resolution NMR spectra can generate very large amounts of data. In the case of metabonomics, a raw NMR spectrum contains as many as 25,000 metabolites (treated as variables here). As in this research it is required to establish possible relationships (or correlations) among the various subjects or variables, the greater the amount of information there is to analyse, the higher the difficulty and complexity of obtaining the required results will be. The original epilepsy data set, as described in Chapters 2 and 4, contains 338 variables for 122 patients, as the bin width size has been set to the quite large value of 0.04 p.p.m., therefore it can be seen that at higher resolutions (with many more metabolites being introduced into the problem, corresponding to decreasing bin sizes), it would be almost impossible to properly examine and analyse the data. Hence, it is necessary to apply suitable statistical methods to increase the chance of identifying any potential similarities or differences among the various samples in the data, by reducing the dimensionality of the input space of the data to a small number of dimensions (usually 2 or 3, as only then can the results of the pattern recognition analyses of the data be graphically depicted).

To classify the samples into groups of similar characteristics, which can give an insight in the situation under investigation, statistical methods such as *Principal Components Analysis* (PCA) and *Cluster Analysis* can be used. Samples classified in a group will have similar characteristics, but different from those in other groups. No information about the groups is known beforehand and no assumptions are necessary concerning the group to which a sample may be classified. These unsupervised pattern recognition techniques aim to facilitate the use of various algorithms in order to reduce the amount of data complexity and afterwards present in a graphical form the patterns or clusters identified in the data.

PCA, the unsupervised technique most commonly applied to metabonomics data (and in general in chemometrics studies) for the reduction of the dimensionality of the data, is reviewed in Chapter 5. The theoretical background of this unsupervised technique is given in Section 5.2, and the application of PCA to the epilepsy data is described in Section 5.3.

Chapter 6 covers *multidimensional scaling* (MDS) techniques (also known as principal coordinates analysis). More specifically, two metric scaling algorithms are reviewed, i.e.

*classical scaling* and *Sammon's nonlinear mapping* (NLM). *Classical scaling* is described in Section 6.2. *Metric* MDS including *Sammon's nonlinear mapping* technique, are covered in Section 6.3. The application of *classical scaling* and *Sammon's nonlinear mapping* to the epilepsy data and their results can be found in Section 6.4.

Several *clustering* techniques are reviewed in Chapter 7. Proximity measures are the subject of Section 7.3. *Hierarchical clustering* methods are covered in Section 7.5 with the main emphasis on *agglomerative nesting* algorithms. Two categories of *optimal partitioning* methods, *fuzzy* and *hard clustering* algorithms, are reviewed in Section 7.6, with the *fanny* and *k*-means algorithms respectively applied to the epilepsy data. *Competitive learning* algorithms are given in Section 7.7 with emphasis on the *self-organizing maps* (SOM) statistical approach and its application to the epilepsy data.

All the above mentioned techniques have been applied to the metabonomics data to assess their capability of reducing the dimensionality of the input space or clustering the data effectively and efficiently and identifying differences between responders and non-responders to AEDs.

# Chapter 5

## Principal Components Analysis

### 5.1 Introduction

Principal components analysis (PCA) is the main tool used by analysts for data reduction. This technique involves the construction of a new set of variables as linear combinations of the original variables in the data set. More specifically, PCA is a statistical technique which aims to reduce the dimensionality,  $n$ , of a data space (Diamantaras and Kung, 1996). It might be possible to describe the data and examine the underlying structure of its variance, by using a smaller number,  $m$ , of independent variables. This intrinsic dimensionality,  $m$ , of the data, depends directly on the correlation between the original (observed) variables. The higher the correlation is, the smaller the number of independent variables that will be needed. The  $n$  observed variables can then be represented as functions of the  $m$  independent variables, called *components*, with  $m < n$ , without losing an important amount of the total variation of the data. Very important also is the possibility of using the extracted components in multivariate calibration of the data. Calibration can be considered as the study of potential quantitative relationships between two or more variables in the spectral data. This is usually achieved by studying how a number of *independent* or *response* variables vary as a function of a *dependent* variable. The techniques used in such studies belong to an area of statistics known as *regression analysis*. In PCA, a common regression technique for calibration is called *principal components regression* (PCR). In this regression method, instead of using the observed variables in multivariate regression (hard modelling), the components can be used as the independent variables, thus relating them to the various concentrations of the metabolites in the data (soft modelling).

PCA is a very popular unsupervised technique in metabonomics, and has been used extensively for extracting the most relevant descriptors of the data, or to reduce the dimensionality of the input space. There are many studies of "omics" profiles of organic samples. Lindon et al. (2001) give a comprehensive description of the use of PCA in bionomics studies, with emphasis to metabonomics and metabolic profiles of organic

samples. An important area of metabolic profiling is toxicology and drug development. Keun (2006) illustrates how PCA of metabolic profiles can help in the detection of drug toxicity-specific biomarkers, as well as how PCA can be used as a projection method in metabonomics toxicology. Another area of interest is the use of PCA to study the effects of various physiological conditions to the metabolic composition of biofluids of organic samples, such as the effects of inter-animal and diurnal variation, gender, age, diet, species, strain, hormonal status and stress on the metabolic profiles of urine of laboratory animals over a given time-course (Bollard et al., 2005b), and the biochemical effects of a diet with isoflavones of premenopausal women on their urine metabonomic profiles (Solanky et al., 2005).

## 5.2 Theoretical Background

The simpler functional form of representation is a linear transformation (combination). The general transformation needed for an  $n$  dimensional space can be written as

$$\begin{aligned} f_1 &= a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \\ f_2 &= a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \\ &\vdots \\ f_n &= a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n \end{aligned}$$

or in a matrix form as  $F = AX$ , where  $F$  is the  $n$ -dimensional component column vector  $(f_1 \ f_2 \ \dots \ f_n)^T$ ,  $X$  is the  $n$ -dimensional column vector  $(x_1 \ x_2 \ \dots \ x_n)^T$  and  $A$  the  $(n \times n)$  matrix of coefficients  $a_{ij}$ ,

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix}.$$

Geometrically, the reduction of the dimensionality of the data space can be seen as the projection of the vector  $X$  onto an  $m$  dimensional space. Usually this space is a line, a plane or a 3-dimensional space to make it possible to represent the data graphically and describe the correlation between the variables.

From the transformation equations, it can be seen that for the estimation of the components, the elements of matrix  $A$  must be calculated. There are two factors of variation in the transformation procedure, namely the variation due to the reduction of the dimensionality of the data space (projection error) and the variation of each component. As each component is represented graphically by a line in a specific direction,

the projection error is the variation around the line, whereas the component's variation is the spread of the data along its line. We want to maximize the component variation, while at the same time minimize the projection error. Using the covariance matrix or the correlation matrix of the variables, we can estimate the eigenvectors and eigenvalues of these matrices (Diamantaras and Kung, 1996; Massart et al., 1990). The estimated eigenvectors are the columns of matrix  $A$  and hence their eigenvalues are the loadings of the components on the observed variables. If we write matrix  $A$  as

$$A = (\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n)$$

where  $\mathbf{a}_i$  is the column vector with elements  $(a_{i1}, a_{i2}, \dots, a_{in})^T$ ,  $i = 1, \dots, n$ , we can then estimate each component  $f_i$  by the column vector  $\mathbf{a}_i$ . The components' covariance matrix,  $C_f$ , can be written in terms of the observed variables' covariance matrix,  $C_x$ , as

$$C_f = AC_xA^T. \quad (5.2.1)$$

As the components are independent, their covariance matrix is diagonal with elements given by the computed eigenvalues,  $\lambda_i$ . From equation (5.2.1), we can derive for each vector  $\mathbf{a}_i$  of matrix  $A$  the following equivalent expression

$$C_x\mathbf{a}_i = \lambda_i\mathbf{a}_i, \quad (5.2.2)$$

where  $\lambda_i$  is the eigenvalue for component  $f_i$  (Diamantaras and Kung, 1996). As it is necessary to compute the components in decreasing order of variation, the first component will have the maximum variance,  $Var(f_1)$ . This is calculated from equation (5.2.2) as the maximum eigenvalue,  $\lambda_1$ . The eigenvector corresponding to this eigenvalue,  $\mathbf{a}_1$ , provides the direction of the first component axis, on which the data is projected. The spread of the projected data on this axis is given by  $\lambda_1$ . Solving equation (5.2.2) for the second largest component variation,  $Var(f_2)$ , we obtain  $\mathbf{a}_2$  and  $\lambda_2$ . The eigenvectors are taken to be orthogonal to each other, i.e.  $\mathbf{a}_i^T\mathbf{a}_j = 0$  for every  $i$  and  $j$ . We continue the procedure until the last component  $f_n$  is found. It is important to note that the total component variation is equal to the variation of the observed variables, that is

$$tr(C_x) = \sum_{i=1}^n \lambda_i. \quad (5.2.3)$$

In addition, the proportion of the total variation that a component  $f_i$  explains is given by

$$\frac{\lambda_i}{\sum_{i=1}^n \lambda_i}. \quad (5.2.4)$$

### 5.2.1 Testing Data Suitability for PCA

An important consideration before accepting any results obtained by PCA, is to assess the amount of information that the data contains. If the amount of information is very large, that is, the descriptors in the data set are not correlated, then there is no point in applying PCA, as there will be no significant data reduction. Two statistics which can confirm the suitability or not of the data for PCA, are the *normalized entropy*,  $\tilde{H}$ , of a data set (Cangelosi and Goriely, 2007) and the *Gleason - Staelin* statistic (Jackson, 2003). The former is given by equation (5.2.5):

$$\tilde{H} = -\frac{1}{\log_2 N} \sum_{i=1}^N p_i \log_2 p_i \quad (5.2.5)$$

where  $p_i$  is the proportion of total variation explained by component  $i$ , and  $N$  is the number of components that PCA calculated (the rank of the  $X$  data matrix).  $\tilde{H}$  takes values in the range  $[0, 1]$ . The higher its value is, the more information is contained and the less useful PCA is. For value 1, all variables in the data set are completely uncorrelated, hence the data space dimensionality is equal to the number of the variables in the data set, whereas if its value is 0, then all variables are completely correlated, and the dimensionality is 1, as only one component is needed to describe all the information. The *information dimension*, related to the *normalized entropy*, and defined as

$$n_0 = \prod_{i=1}^N p_i^{-p_i} \quad (5.2.6)$$

where  $p_i$  is the proportion of total variation explained by component  $i$ , and  $N$  is the number of components that PCA calculated, can be used to measure the number of components to retain.

The *Gleason - Staelin* statistic is given by equation (5.2.7):

$$\phi = \sqrt{\frac{\|R\|^2 - n}{n(n-1)}} \quad (5.2.7)$$

for the correlation matrix  $R$ , where  $n$  is the dimensionality of the data set - the number of original variables in the data - and

$$\|R\|^2 = \sum_{i=1}^n \sum_{j=1}^n r_{ij}^2$$

with  $r_{ij}$  being the correlation between variables  $i$  and  $j$ . The statistic becomes

$$\phi = \sqrt{\frac{\|S\|^2 - \sum_{i=1}^n (s_i^2)^2}{\sum_{i=1}^n \sum_{j \neq i}^n (s_i s_j)^2}}$$

when the covariance matrix  $S$  is used. Similarly to the normalised entropy,  $\phi$  takes values in the range  $[0, 1]$ . In this case, though, the higher its value is, the more correlated the variables in the dataset are. For value 0, the variables are totally uncorrelated, and there is no point in applying PCA to the data, whereas if its value is 1, then there is perfect correlation among the variables and the dimensionality of the data space is 1.

## 5.2.2 Determining the Number of Components to Extract

A very important part of the PCA procedure is the identification of the maximum number of required PCs. There is a long dispute in the literature on which method of estimation of PCs is the most appropriate, however none of the suggested ways of tackling this problem is suitable for every possible situation. There is a large number of stopping rules. They can be divided into categories, such as those rules which are based on confidence intervals, e.g. *parallel analysis* and *re-sampling* methods (Horn, 1965; Besse and Falguerolles, 1993) and those based on average test statistic values e.g. *broken stick* and *Velicer's MAP* (Neto et al., 2005; Ferre, 1995; Velicer, 1976). Two commonly used stopping rules, one based on average test statistic value and one based on confidence intervals, are described below.

- **Broken-stick.** This is based on the concept that by dividing randomly the total variance of a multivariate data set, the distribution of the eigenvalues follows a broken-stick distribution. The idea is that if a line segment is randomly divided into  $n$  pieces, then the expected value of the length of the  $k^{th}$  piece is given by

$$E_k = \frac{1}{n} \sum_{x=k}^n \frac{1}{x}. \quad (5.2.8)$$

If the eigenvalue of the component  $k$  is larger than the respective expected value,  $E_k$ , of the broken-stick distribution, then this component is retained (Cangelosi and Goriely, 2007; Neto et al., 2005; Legendre and Legendre, 1998). However, use of this stopping rule requires a bit of caution, as according to Cangelosi and Goriely (2007) it sometimes underestimates the appropriate number of principal components. Comparison of the result of this criterion with those of other stopping rules is a wise precaution to avoid retaining fewer than the appropriate number of principal components.

- **Parallel Analysis.** This stopping rule was introduced by J.L. Horn (Horn, 1965). It is based on the generation of random data sets of uncorrelated normally distributed variables of the same size as the original data. The method proceeds by applying PCA to these data sets and retaining the eigenvalues for each principal component. This is repeated for a large number of times, e.g. 1000. The percentile intervals of eigenvalues for each component are calculated, for instance, at significance level 95%. If the observed values exceed those of the calculated intervals, then the null hypothesis at the chosen level of significance is not accepted, therefore the component is retained. It should be noted that as this analysis depends on the normality of the generated data, it may not be the most suitable when the generated data is not normally distributed. In such cases, non-parametric re-sampling techniques such as the *bootstrap* methods may give more robust results (Daniel, 1992; Besse, 1992).

These two methods will be used in the analyses, and their results in the epilepsy data will be compared to some other popular rules, such as *Kattel's Scree Test* and the number of components explaining 90%, 95% and 99% of the total variance in the data.

## 5.3 Application of PCA in the Epilepsy Data

### 5.3.1 Introduction

The generation and pre-processing of the NMR signals (spectra) were carried out by a NMR scientist, Dr. Parkinson, from the Department of Pure and Applied Chemistry at the University of Strathclyde. More specifically,  $^1\text{H}$  NMR spectroscopy was performed in a dedicated facility at the Department of Chemistry, University of Edinburgh, UK. A 50  $\mu\text{l}$  aliquot of deuterium oxide ( $D_2O$ ; 99.9 %  $^2\text{H}$  atom; Sigma, UK) was added to 500  $\mu\text{l}$  of serum. Particulates were removed by microcentrifugation for 1 minute at 10,000 rpm and the supernatant transferred to a 5 mm precision NMR sample tube (Wilmad 528-PP-7; Aldrich, UK). One dimensional proton NMR spectra were acquired using a Bruker Avance 600 NMR spectrometer equipped with a triple resonance TXI [ $^1\text{H}$ ,  $^{13}\text{C}$ ,  $^{15}\text{N}$ ]-xyz triple axis gradient probe-head operating at a proton resonance frequency of 599.813 MHz. Data were acquired on non-spinning samples at a temperature of 25 $^\circ$  C using a noesyprsat pulse sequence (d1-p1-d2-p1-t $_m$ -p1-acq, where  $d1 = 2.0$  s,  $p1 = 12.75$   $\mu\text{s}$  [90 $^\circ$  r.f. pulse at a relative power level of -2.5 dB],  $t_m = 50$  ms and  $acq = 1.639$  s). Data were acquired with 64 transients over a frequency bandwidth of 10 kHz, digitised into 32,768 data points, Fourier-transformed without any applied weighting function, phase and baseline corrected, and referenced internally to the methyl doublet resonance of lactate at 1.33 ppm (parts per million). All  $^1\text{H}$  NMR data were processed remotely using Xwin-NMR (version 3.5) and subsequently read directly into the bucketing and statistics



module of AMIX (version 3.6.6; Bruker Biospin, Germany) which allows for data scaling and integration, alternative bucketing, inclusion/exclusion of individual datasets, and presentation of analyses according to different combinations of principal components. Data were divided into 0.04 ppm buckets over a spectral range of 0.0 – 6.0 ppm (Zweiri et al., 2010).

As described previously in Sections 2.4.2, 2.4.3 and 5.2.2, the original epilepsy data consist of 97 patients who are either responders or non-responders to AEDs treatment. As is usual in proton NMR metabonomics data, the recorded NMR spectra include resonances which do not correspond to any endogenous metabolites (Ross et al., 2007). Spectral regions which do not contain any endogenous metabolites are not useful in data analysis and therefore need to be removed before any data analysis is performed. Such regions are the spectral ranges below 0 ppm and above 10 ppm (Williams and Fleming, 1995). In addition, the spectrum resonances in the spectral range 4.7 – 4.9 ppm, which are the remaining water resonances after the application of water suppression techniques in the spectra, need to be excluded as well. The reason for this fact is that the analysis of signals of metabolites below the water resonances is not possible as the water peak dominates the proton NMR spectrum, thus, affecting the multivariate data analysis of the spectral peaks of interest (Ross et al., 2007). After the exclusion of these regions from the spectral data, the remaining spectral data consist of 97 subjects and 244 variables in the spectral range 0.02 – 9.98 ppm. As the variables in the spectral range 6.02 – 9.98 ppm were very low in intensity (Zweiri et al., 2010), these were also excluded, with the remaining spectral data containing 97 patients and the 144 variables in the spectral range 0.02 – 5.98 ppm. Finally, each and every sample in the data set was subjected to row-scaling to a constant total, to make the spectra more comparable. This was done by dividing each variable by the sum of all variables in a sample, effectively converting the absolute intensities of the data to proportions.

The possibility of reducing the number of variables to a far smaller number of components, without losing any important information from the original data, will be examined. Applying PCA to the data may also indicate any relationships among the samples and the variables. Any potential clusters may be identified on the resulting plots. After identifying the required PCs for the variables in the data set, it will be established using appropriate criteria (statistics) whether more than 2 or 3 components are needed, to describe enough of the variation in the data. These PCs will describe with high accuracy most of the important information in the original data, possibly facilitating the identification of existing clusters of patients and/or metabolites. In addition, the patients' clinical characteristics will be investigated in order to clarify if there is any relationship between them and the patients' metabolic profiles. That is, it will be assessed whether PCA can identify any natural clusters of patients with respect to their clinical characteristics and more importantly to their *response to AEDs*.

### 5.3.2 Data Suitability

Before doing any PCA analysis, it is necessary to test the suitability of the data for PCA. The normalised entropy and the Gleason - Staelin statistic will be calculated using equations (5.2.5) and (5.2.7) respectively. The *normalised entropy* for the epilepsy data set is 0.121, which means that the metabolites are highly correlated, with the dimensionality of the data being close to 2 (the value of the *information dimension* is  $1.74 \approx 2$ ). In addition, the value of the *Gleason - Staelin* statistic using the correlation matrix is 0.555, indicating that the metabolites are sufficiently correlated to justify data reduction using techniques such as PCA. Both statistics confirm beyond any doubt that the data is suitable for PCA analysis.

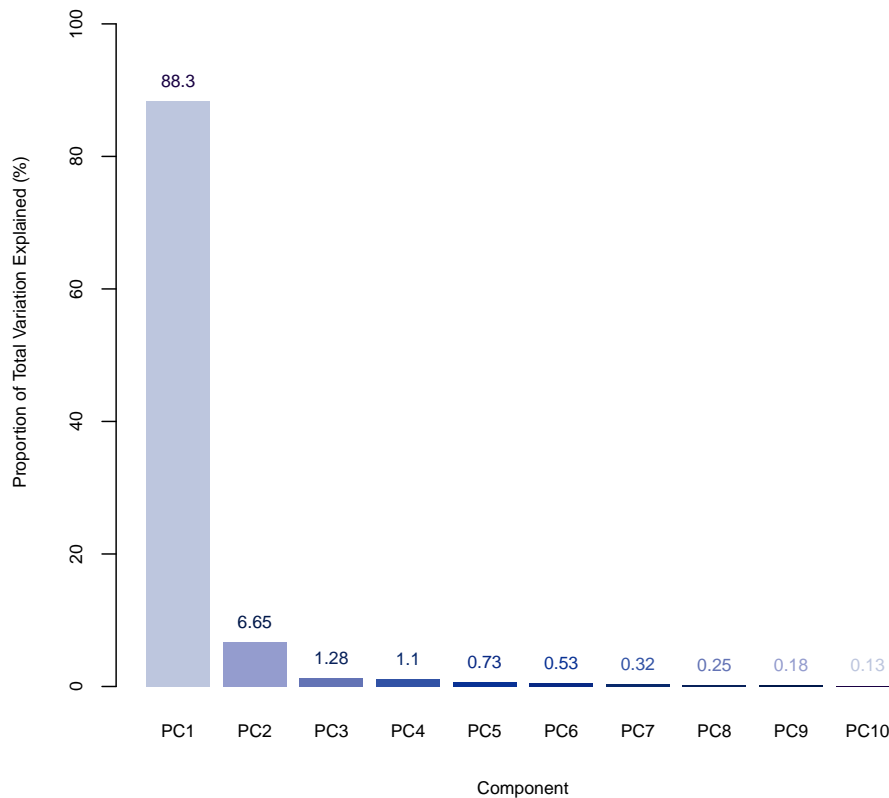
### 5.3.3 Identification of the Number of Components to Retain

After confirming the suitability of the data, the next step in PCA is to identify the number of principal components to retain for the analyses. The percentages of the total variation in the data explained by the first ten principal components can be seen in Figure 5.1. The plot shows that about 90%, 95% and 99% of the total variation is explained by 2, 3 and 8 PCs respectively. Table 5.1 contains the standard deviation, the percentages of the total variance explained and the cumulative percentages of variance for the first ten PCs. The detailed results for the variance of the PCs indicate that no more than 2 components need to be retained for further analyses, as they explain most of the variation in the data,  $\approx 95\%$ , while the variation of the remaining components is likely to be due to measurement and instrumentation errors. To confirm these findings, as described in Section 5.2.2, the *broken stick* and *parallel analysis* stopping rules will be used to identify the appropriate number of principal components.

An illustration of the broken-stick model can be seen in Figure 5.2 (top), showing that only 2 components should be retained, as only two eigenvalues are larger than the expected values of the broken-stick distribution (red line). *Cattell's scree test* (Cattell, 1966) is also depicted in Figure 5.2 (black line in the top figure), confirming that at most 3 components should be retained (using one more component after the break in the line (Jackson, 2003)).

**Table 5.1:** Standard deviation, percentage of total variance explained by, and cumulative percentages of variance for the first ten PCs.

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10
Standard Deviation	0.0150	0.0041	0.0018	0.0016	0.0013	0.0011	0.0009	0.0008	0.0007	0.0006
Proportion of Variance (%)	88.30	6.65	1.28	1.10	0.73	0.53	0.32	0.25	0.18	0.13
Cumulative Proportion (%)	88.30	94.95	96.23	97.33	98.06	98.59	98.91	99.16	99.34	99.47



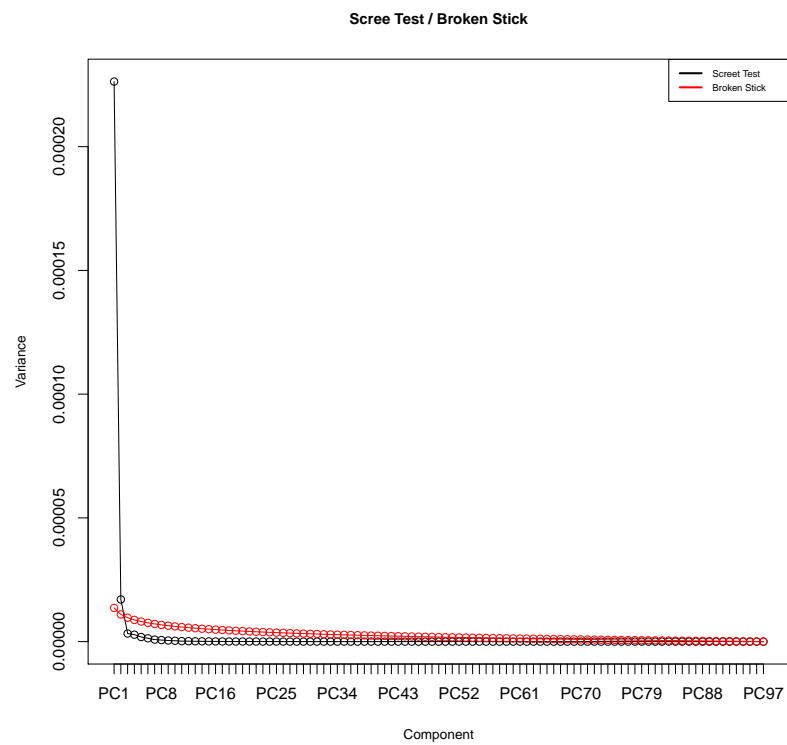
**Figure 5.1:** Percentages of the total variation in the data explained by the first ten components.

Parallel analysis was performed using the mean and the 99<sup>th</sup> centile estimates for the calculation of the confidence intervals, and different numbers of random sets of up to 200 per variable. All runs retained 1 component, independently of the confidence intervals and number of iterations used. The parallel analysis plot in Figure 5.2 (bottom) illustrates the adjusted and unadjusted eigenvalues and suggests that 1 component should be retained. The unadjusted eigenvalues are the eigenvalues of the observed data from an unrotated PCA. The random eigenvalues are the estimated, either mean or centile, eigenvalues from 4320 iterations, which is the default number of iterations, given by  $30 * \text{number of variables}$ , as used by the R function `paran()` to perform parallel analysis. The adjusted eigenvalues are given by the adjustment

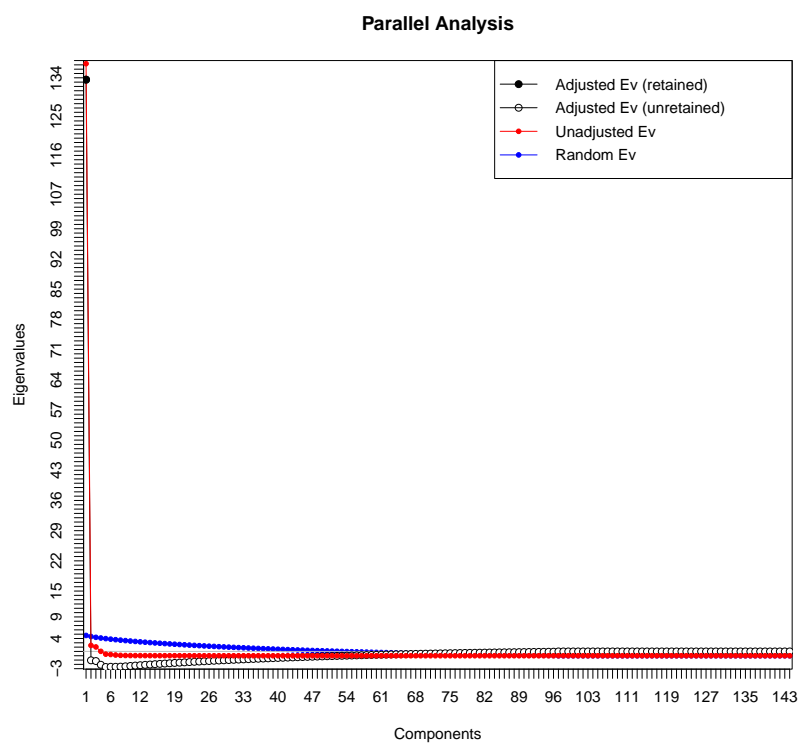
$$\text{AdjustedEig} = \text{UnadjustedEig} - (\text{SimulatedEig} - 1)$$

and retained if their values are greater than 1.

In Table 5.2, a comparison of the results for a number of stopping rules can be seen. The results stated in Table 5.2 show that 2 or 3 PCs should be retained. The result of retaining 1 PC, from *parallel analysis*, although being the smallest, if chosen will not be particularly interesting, and probably important information contained in the



(a) Scree and Broken Stick plots



(b) Parallel Analysis plot

Figure 5.2: Stopping rules for the number of components.

**Table 5.2:** Comparison of various stopping rules for the selected epilepsy data of 97 patients and 144 variables.

Stopping rule	Number of Components retained
Parallel Analysis	1
Brocken Stick	2
Kattel's Scree Test	3
90% of Variance	2
95% of Variance	3
99% of Variance	8
Information Dimension	2

second PC will not be considered. Therefore, despite the first component explaining approximately 88.3% of the total variation, one PC is most probably not the appropriate number of PCs to retain. Retaining the first two or three principal components allows for proper graphical representation of the data and easier identification of any natural patterns in the structure of the input space defined by the selected data.

### 5.3.4 Results of PCA

Having identified that the first two or three PCs should be retained for further analyses, graphical representation of the data structure is the next step in the PCA analysis. The PC scores (concerning the samples) and loadings (concerning the variables) can be plotted in many ways to give a visual summary of the epilepsy data. These can be in 1, 2 or 3 dimensions.

Plotting the PC scores is usually the first step in describing the data graphically. The 1-dimensional scores plot is essentially a bar chart, where, for a selected PC, each score is plotted against sample number. It is often useful to re-order the sample in ways that can facilitate better the interpretation of the scores. Selecting a suitable order of the samples should indicate clearly in the bar chart if a specific PC is influenced by a specific grouping of the samples. One way of indicating a particular grouping of the patients is by using colour. In the case of the epilepsy data, the groupings of the patients will be defined by their clinical characteristics of interest (*Gender, Seizure Type, Response to AEDs, Age and BMI*). In a 2-dimensional scores plot, the scores of a PC are plotted against those of another. This is usually done for the first 3-4 PCs, which more often than not are sufficient to explain most of the variation in the data. In this case, the samples are plotted using the values of the scores as coordinates. This type of plot may indicate which of the PCs appears to be the best discriminator for a specific grouping of the patients. The groupings are usually represented by a different symbol and/or colour. In the case of the epilepsy data, whenever 2-dimensional score plots are used, different colours will represent the groupings of patients according to

their clinical characteristics. Finally, if the results of the 1 and 2-dimensional plots are not conclusive, 3-dimensional scores plots can be used, such that each axis of the plot represents one PC. Colouring of the samples can be applied in an analogous way to that of the 1 and 2-dimensional plots. In the epilepsy data, 3-dimensional plots will be used only if the results of the lower dimensional plots justify it.

A general visual summary of the epilepsy data can be seen in Figure 5.3. The most interesting plots of the six are the scores plot for the first two components and that of the pair PC3 and PC4. The former scores plot describes most of the information in the epilepsy data (approximately 95% of the total variation is explained by the first two PCs as shown in Table 5.1), therefore is necessary to investigate these two PCs. The latter score plot indicates that there is one patient with ID number 44 having a high positive score in PC4, which seems to be an outlier, influencing mainly PC4. The patient with ID number 20 has a very high negative score in PC1 and a high positive score in PC3, and could influence PC1 and PC3. As PCA is affected by outliers, it is important to confirm whether these samples are outliers or not.

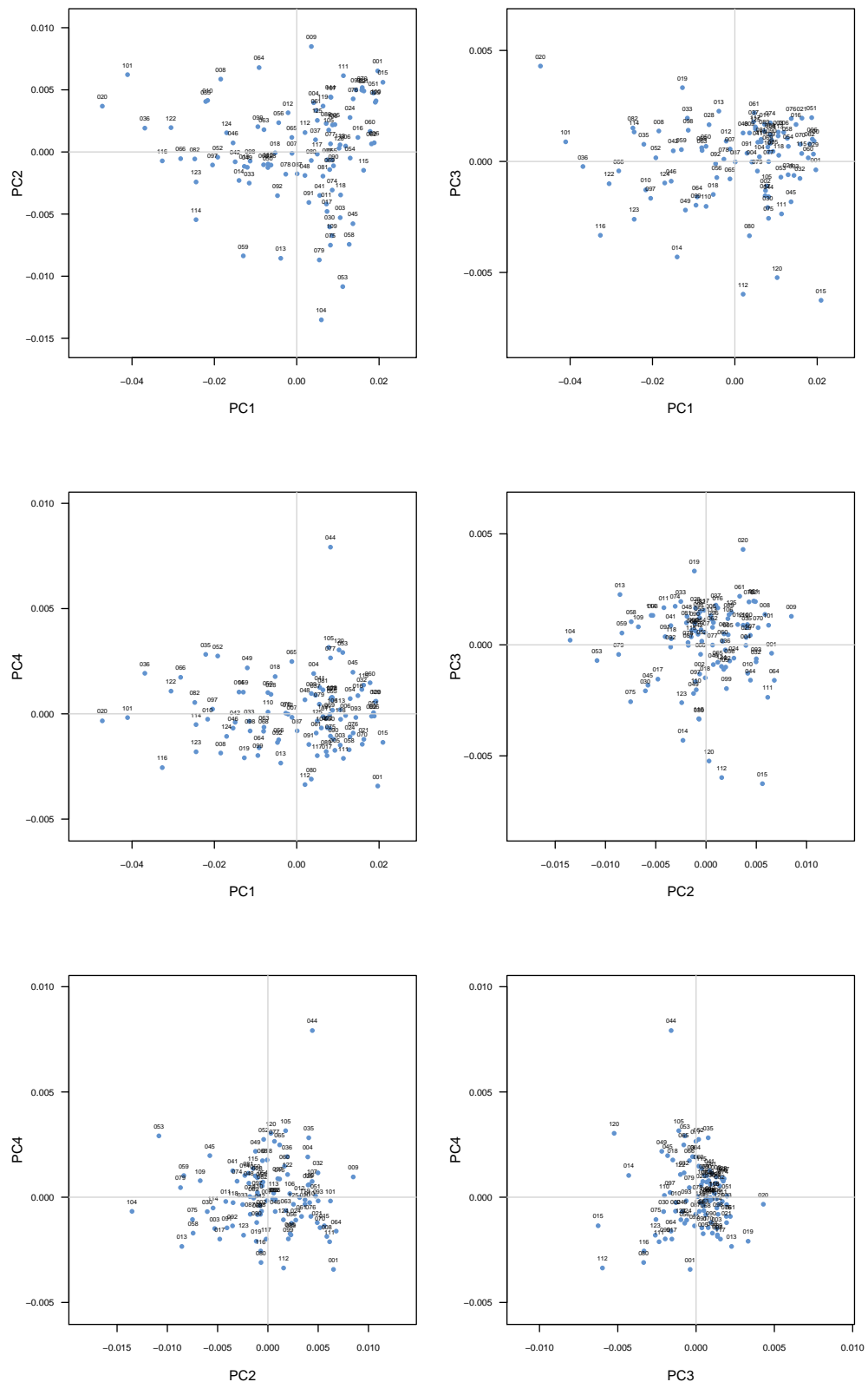
There are two types of outliers that could affect PCA, *orthogonal* outliers and *leverage points*. The former are related to their *orthogonal distance* to the space defined by the PCs, and the latter to their *score distance*, their projection's distance from the centre of the PCA space. The *score distance*,  $SD$ , of a sample  $i$  is given by

$$SD_i = \left[ \sum_{k=1}^{N_{pc}} \frac{t_{ik}^2}{v_k} \right]^{\frac{1}{2}}$$

where  $N_{pc}$  is the number of PCs forming the PCA space,  $t_{ik}$  the elements of the score matrix and  $v_k$  the variance of the  $k^{th}$  PC (Varmuza and Filzmoser, 2009). Assuming that the data is multivariate normally distributed, the squared score distances can be approximated by a chi-square distribution,  $\chi_{N_{pc}}^2$ , with  $N_{pc}$  degrees of freedom. A cutoff value for the score distance can be the 97.5% quantile,  $\sqrt{\chi_{N_{pc},0.975}^2}$ . If the score distance of a sample is larger than this cutoff point, then the sample is a leverage point. The *orthogonal distance*,  $OD$  of a sample  $i$  is defined as

$$OD_i = \|x_i - Pt_i^T\|$$

where  $x_i$  is the  $i^{th}$  sample of the centred data matrix,  $P$  the loadings matrix using  $N_{pc}$  PCs and  $t_i^T$  the transposed score vector of sample  $i$  for  $N_{pc}$  PCs (Varmuza and Filzmoser, 2009). A cutoff value for the orthogonal distance is computed by Hubert et al. (2005), using the Wilson-Hilferty approximation for a chi-square distribution. That is, the distribution of  $OD^{\frac{2}{3}}$  is approximately normal, with the centre (mean) and spread (variance) of the values being robustly estimated, e.g. using the median and the median absolute deviation (MAD) respectively. The cutoff value is then computed as



**Figure 5.3:** Scores plots for the epilepsy data for the first four PCs, superimposed with the patient ID numbers. The sample numbers in the plots are the original ID numbers of the selected 97 patients. The data are row-scaled to a constant total.

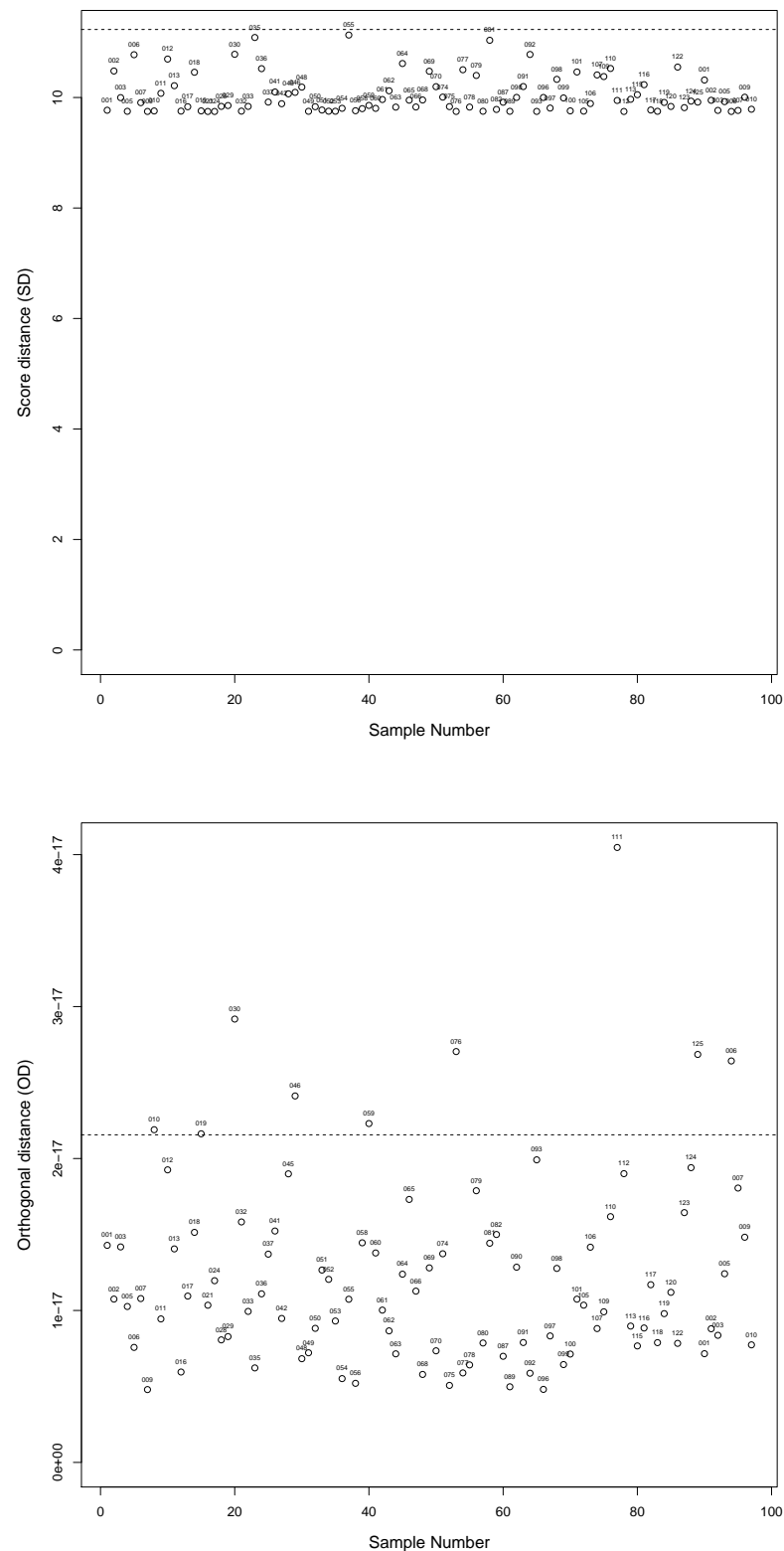
$(\text{median}(OD^{\frac{2}{3}}) + MAD(OD^{\frac{2}{3}})z_{0.975})^{\frac{3}{2}}$  where  $z_{0.975}$  is the 97.5% quantile of the standard normal distribution. If the orthogonal distance of a sample is higher than the cutoff value, then the sample is an *orthogonal outlier*. If an orthogonal outlier sample with large orthogonal distance also has a large score distance (so it is a leverage point), then the sample is a *bad-leverage* point, as it can affect negatively the correct estimation of the PCA space. A leverage point that also has a small orthogonal distance but still is an orthogonal outlier, with a large score distance is a *good leverage point*, as it can stabilise the estimation of the PCA space.

Diagnostic plots using the score and the orthogonal distance of the samples in the data can be seen in Figure 5.4. The cutoff values for the *score* and the *orthogonal distance* are equal to 11.23127 and 2.156164e-17 respectively. It can be seen that there are no points with score distance higher than the cutoff value, although patients 35, 55 and 81 have the highest score distances, being close to the cutoff value. In the case of the orthogonal distances, there are 8 points with orthogonal distances higher than the cutoff value, namely the patients 6, 10, 30, 46, 59, 76, 111 and 125, and 1 patient with orthogonal distance approximately equal to the cutoff value (patient 19). However, removing these samples from the data set and re-running the analyses showed that there was no effect from the inclusion of these patients in the PCA, as the results were approximately similar with only an expected reduction of the variance explained by the first PC being slightly smaller (by approximately 5% than that in the total set (Figure 5.5)). Therefore, the original data set of the selected 97 patients can be used for further analyses.

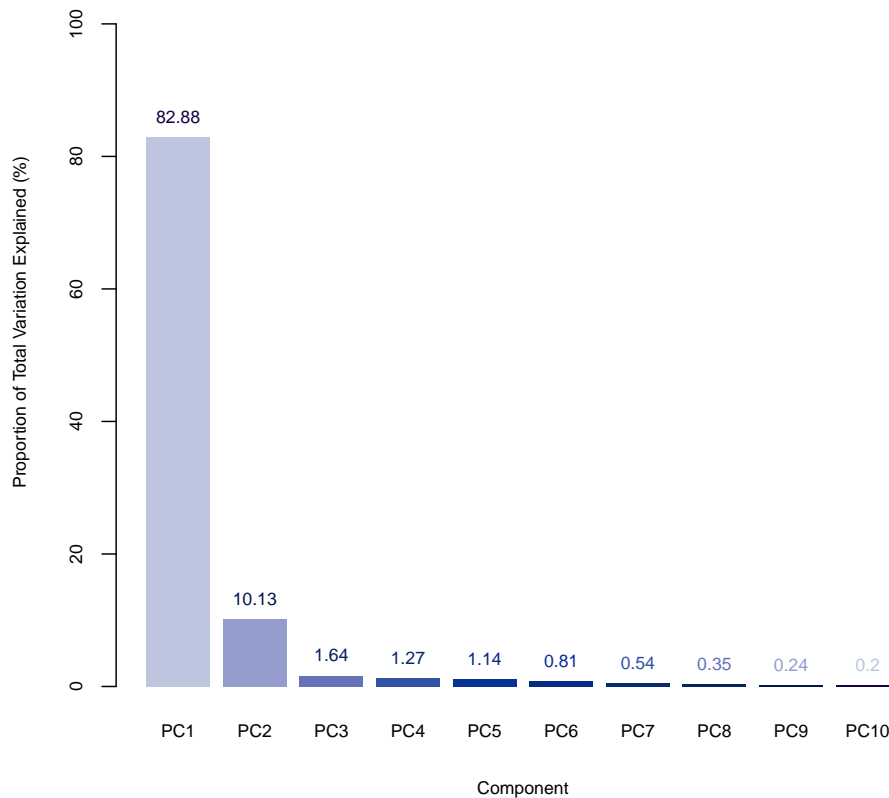
The patients' coordinates with respect to the first four components, which explain approximately 96.23% of the total variation in the data, can be seen in Figure 5.6, superimposed with the *Gender* information of the patients. *Males* and *females* are depicted in red and black colour respectively. From the information about *Gender* in the scores plots for the first four components, it is clear that components PC1 and PC4 cannot distinguish the patients with regards to their *Gender*, whereas PC2 seems capable of discriminating between *females* and *males* with reasonable accuracy. The *female* and *male* patients have mainly low and high scores respectively in PC2. Component PC3 also appears to separate out the *female* patients, with the majority of *females* having high scores. In general, components PC2 and PC3 are related to *Gender*, and especially PC2, as it can discriminate both categories quite satisfactorily. However, the scores on PC3 and PC4 are fairly small compared to those of PC1 and PC2, as these components explain only approximately 2.5% of the total variation of the data. Two PCs are required to separate out the patients with respect to their gender.

Considering the *Seizure Type* of the patients, the scores plots (Figure 5.7) indicate that only *IGE* patients are clearly related to any of the first four PCs. More specifically, it can be seen that in the first three PCs, the *IGE* patients have mainly high scores.



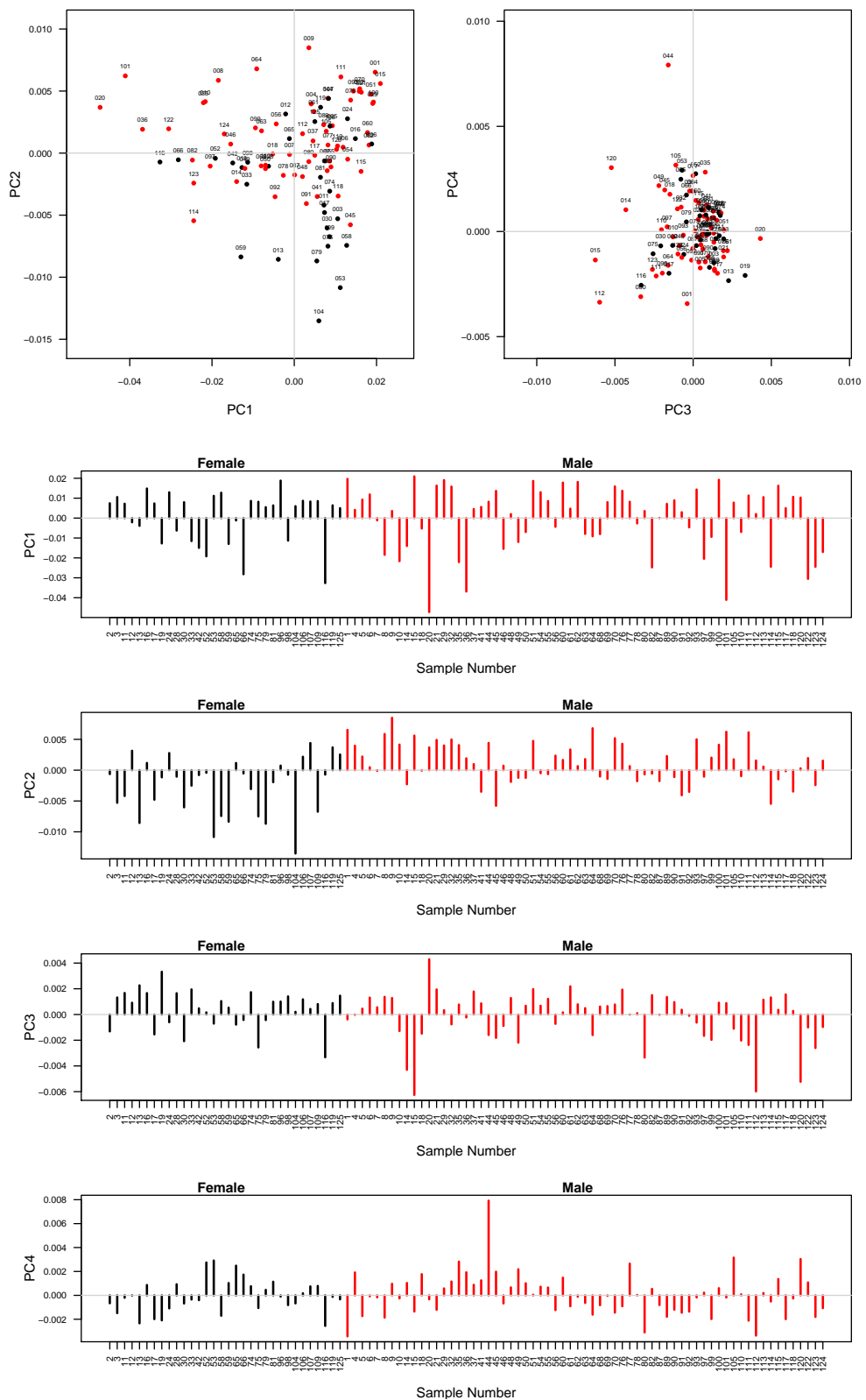


**Figure 5.4:** Outlier diagnostic plots using the score ( $SD$ ) and the orthogonal distance ( $OD$ ). The sample numbers in the plots are the original ID numbers of the selected 97 patients. The horizontal lines in the two plots represent the cutoff values, such that any point above these lines is a *leverage point* (top plot) or an *orthogonal outlier* (bottom plot).

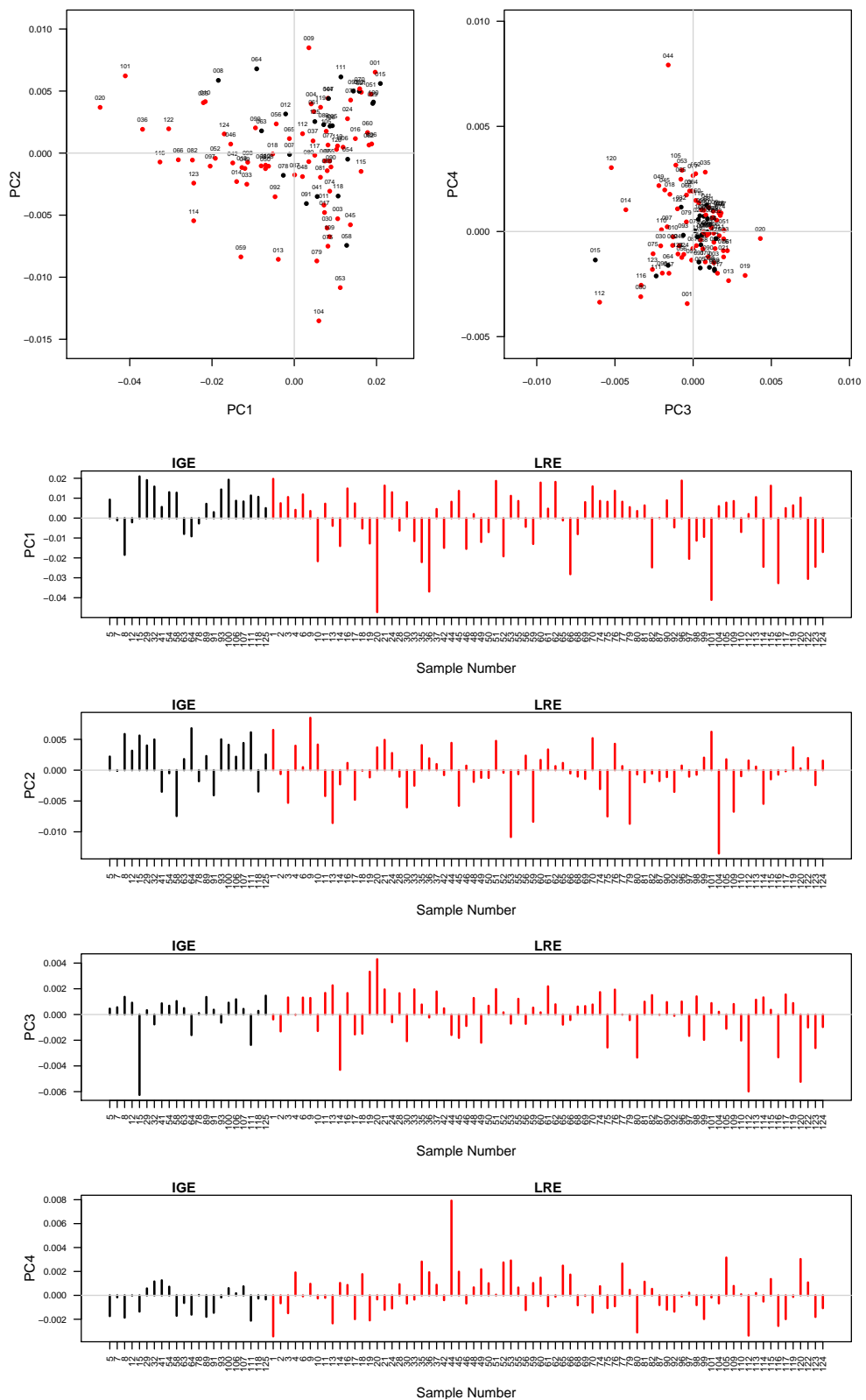


**Figure 5.5:** Percentages of the total variation in the data explained by the first ten components (after removing the outliers).

*LRE* patients 10, 35, 82, 97, 114 and 123 on PC1, and *LRE* patients 53 and 104 on PC2, are all very highly negatively scored. PC4 cannot distinguish the patients with respect to their seizure type. Probably more than four PCs are required to separate out the *Seizure Type* categories. A number of *LRE* patients have very low scores on PC1 and PC2, and *IGE* patients have high scores, but none of the first four PCs can separate out the patients with respect to their *Seizure Type*. It has been demonstrated though, that PC1 is the best PC to separate the *IGE* patients from the others with regards to high values. The scores plot of the first two PCs also indicates clearly that the *IGE* patients tend to lie towards the top-right corner of the plot. To confirm this result, that is, if indeed the first component contributes considerably to the prediction of the *Seizure Type* of the patients, as well as whether the other PCs contribute to the *Seizure Type* at all or not, a *principal components regression* (PCR) analysis can be done. The response variable is the *Seizure Type* information (recoded as binomial with *IGE* patients set to value 0 and *LRE* patients to 1), and the explanatory variables are the first four PCs. The general linear model (glm) with link function  $g$ , the logit



**Figure 5.6:** Scores plots for the epilepsy data superimposed with the *Gender* information. In all plots, *males* and *females* are depicted in red and black respectively. The sample numbers in all scores plots are the original ID numbers of the selected 97 patients. The data are row-scaled to a constant total.



**Figure 5.7:** Scores plots for the epilepsy data superimposed with the *Seizure type* information. In all plots, *IGE* and *LRE* patients are depicted in black and red respectively. The sample numbers in all scores plots are the original ID numbers of the selected 97 patients. The data are row-scaled to a constant total.

function, is given by

$$\text{logit}(\pi) = \log\left(\frac{\pi}{1-\pi}\right) = \beta_0 + \beta_1 PC_1 + \beta_2 PC_2 + \beta_3 PC_3 + \beta_4 PC_4$$

with  $\pi = Pr(\text{Seizure Type} = \text{LRE})$ , where the parameter  $\beta_j$  is associated with explanatory variable  $PC_j$ , such that  $e^{\beta_j}$  is the odds that the response variable takes the value 1 (LRE) when  $PC_j$  increases by one, and (Everitt and Hothorn, 2006)

$$g(\pi) = \beta_0 + \beta_1 PC_1 + \beta_2 PC_2 + \beta_3 PC_3 + \beta_4 PC_4. \quad (5.3.1)$$

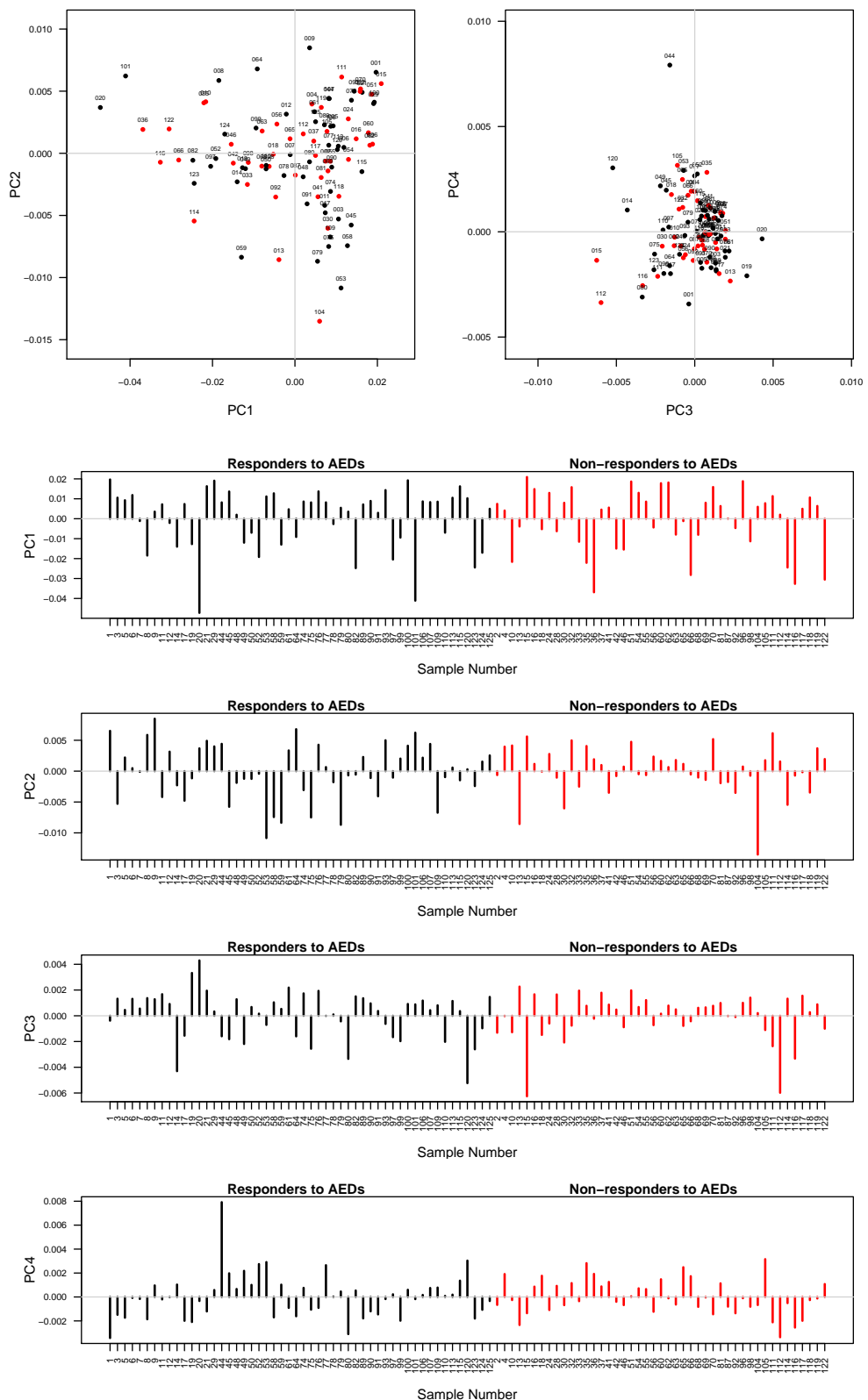
The coefficients of this regression model can be seen in Table 5.3. The PCR results

**Table 5.3:** Coefficients for the PCR model of *Seizure Type* defined by equation 5.3.1 and with a binomial error distribution. The PC scores for the first four PCs have been used as explanatory variables. Results obtained using `glm()` in R with binomial error.

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	1.464	0.292	5.003	5.63e-07
PC1	-43.541	21.127	-2.060	3.93e-02
PC2	-151.275	72.111	-2.097	3.59e-02
PC3	-39.850	144.780	-0.275	7.83e-01
PC4	240.184	176.104	1.363	1.72e-01

show that the first two PCs are significant at 95%, as the  $p$ -values for both these PCs are  $< 0.05$ . The first two components, which explain almost 95% of the total variation in the data, are very important in the analyses. It is clear from the glm results that both these PCs make an evident contribution to the prediction of the *Seizure Type* of the patients. This result confirms the findings of the scores plots for these PCs, as seen previously.

In the case of the *Response to AEDs* information, Figure 5.8 shows that the patients cannot be separated with respect to this characteristic, as none of the first four PCs is capable of distinguishing the patients with regards to their *Response to AEDs* information. The distribution of the scores over the *Response to AEDs* groups is balanced between high and low values. The lowest scored patients on PC1 are 20 and 101 (*responders*) and 36 and 116 (*non-responders*). In PC2, patients 53 and 104 are the lowest scored *responder* and *non-responder* respectively. The first four components do not seem to be sufficient to separate out the patients with respect to their *Response to AEDs* information. To confirm this result, a PCR analysis can be done, with response variable being the *Response to AEDs* information (binomial with values 1 and 0 for a responder and non-responder respectively), and explanatory variables the first four PCs. A similar glm, to that of the *Seizure Type*, can be described as in equation 5.3.1 where in this case,  $\pi = Pr(\text{Response to AEDs} = 1)$ . The coefficients of the regression model can be seen in Table 5.4. The glm results show that there are no PCs that make any significant contribution to the prediction of *Response to AEDs*, as the  $p$ -values for all PCs are larger than 0.05.



**Figure 5.8:** Scores plots for the epilepsy data superimposed with the *Response to AEDs* information. In all plots, *responders* and *non-responders* patients are depicted in black and red respectively. The sample numbers in all scores plots are the original ID numbers of the selected 97 patients. The data are row-scaled to a constant total.

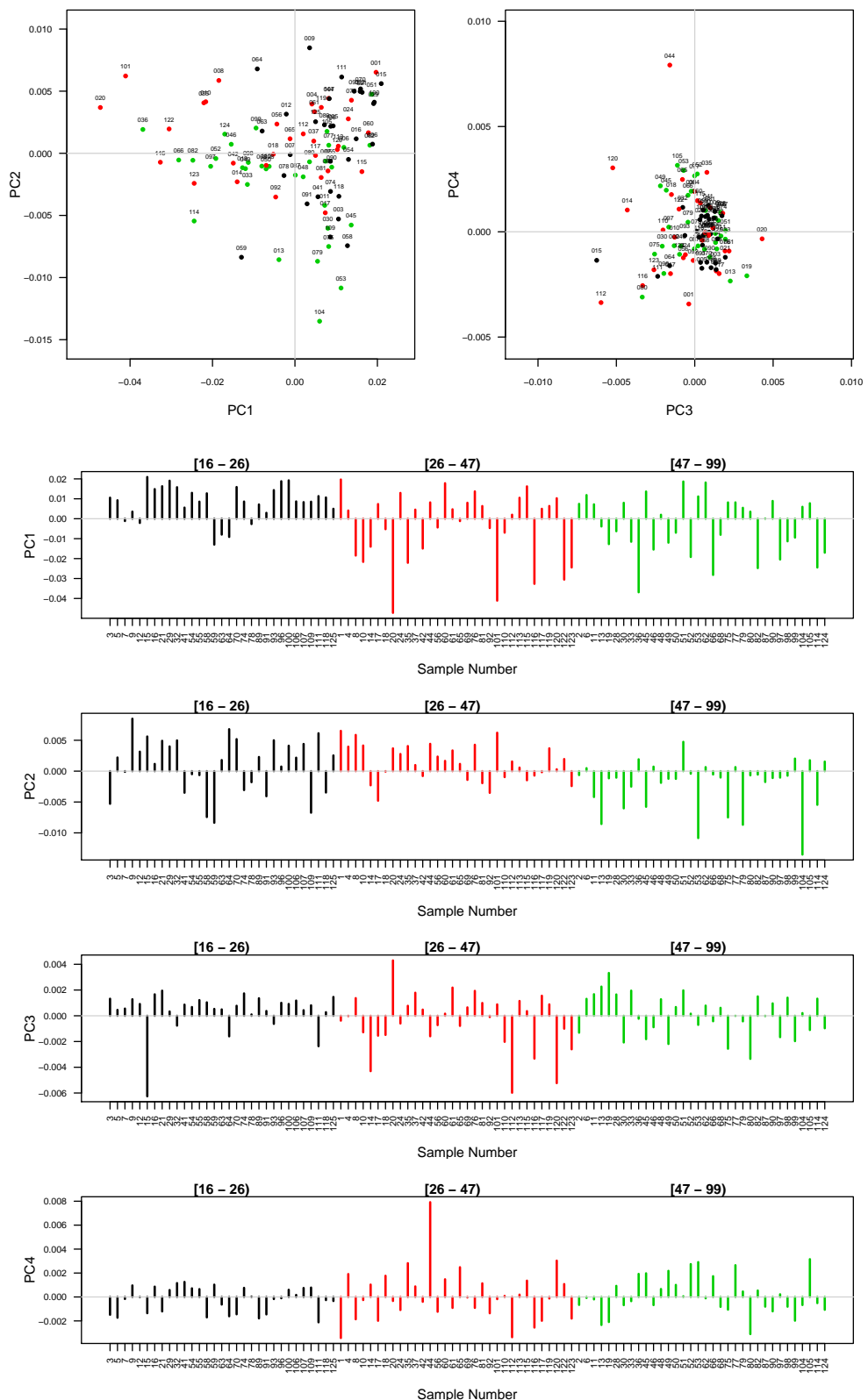
**Table 5.4:** Coefficients for the PCR model of *Response to AEDs* defined by Equation 5.3.1 and with binomial error distribution. The PC scores for the first four PCs have been used as explanatory variables. Results obtained using `glm()` in R with binomial error.

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	0.145	0.204	0.710	0.477
PC1	4.172	13.612	0.306	0.759
PC2	-12.795	49.795	-0.256	0.797
PC3	54.785	113.285	0.483	0.628
PC4	11.391	122.051	0.093	0.925

Figure 5.9 indicates that there is a relationship between the first three components and the age of the patients. More specifically, PC1 and PC3 can separate out the young patients in the *Age* category [16-26), as most patients in this category are highly scored in these components. The remaining two *Age* categories, [26-47) and [47-99), are clearly distinguishable along PC2 according to the size of their scores, with the patients in the former category having high scores and patients in the latter *Age* category having low scores. In general, patients on the last two *Age* categories have much lower scores on PC1. Two or three PCs are sufficient to separate out the patients with respect to *Age*.

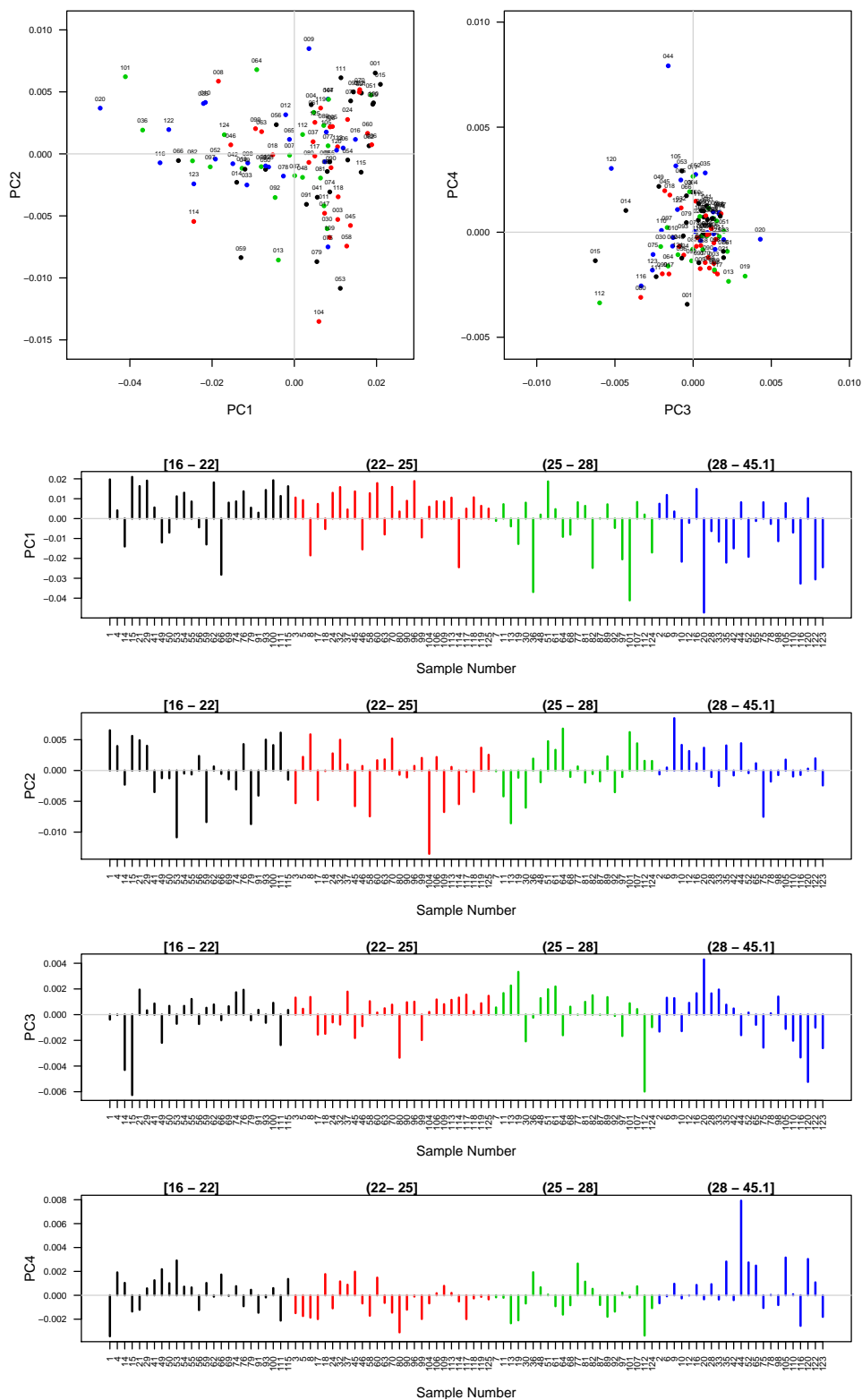
The patients have been divided into four categories with respect to their *BMI* values, namely [16-22], (22-25], (25-28] and (28-45.1]. The scores plots superimposed with the *BMI* information in Figure 5.10 demonstrate that there is a relationship between the first three components and the *BMI* categories of the patients. Patients with *BMI* values in the first two categories (depicted in black and red respectively), are generally separated from the last two categories (depicted in green and blue respectively) on the first component, as the former have high scores and the latter low PC1 scores. The first two categories have similar scores in size in all four PCs, therefore, they cannot be distinguished from each other in these four PCs. The third *BMI* category is related to the third component, with only a few patients in this category having low PC3 scores. The last *BMI* category, (28-45.1], although more difficult to see, seems to be related to PC1 and PC2. In this case, the scores of the patients in PC1 are mainly very low, whereas in PC2 they are mainly high. However, four PCs do not seem to be sufficient to separate out all four *BMI* categories.

*Male* (20, 36, 101 and 122) and *female* (66 and 116) patients have the lowest PC1 scores, while *female* patients 53 and 104 have the lowest PC2 scores. For PC3 *male* patients 15, 112 and 120 have the lowest scores and for PC4 one *male* patient can be easily distinguished among all patients as having the highest score. In general, PC1 is concerned mainly with *IGE* patients, of either gender and response to AEDs, in the *Age* category [16-26) with *BMI* values mainly in the range [16-25]. PC2 is related mainly to *IGE* patients, having either response to AEDs, being of any gender, of age in the range [26-99) and of *BMI* values rather in the range (28-45.1]. The third component is, likewise PC1 and PC2, concerned mainly with *IGE* patients.



**Figure 5.9:** Scores plots for the epilepsy data superimposed with the Age information. In all plots, patients in the three Age categories, [16-26), [26-47) and [47-99) are depicted in black, red and green respectively. The sample numbers in all scores plots are the original ID numbers of the selected 97 patients. The data are row-scaled to a constant total.





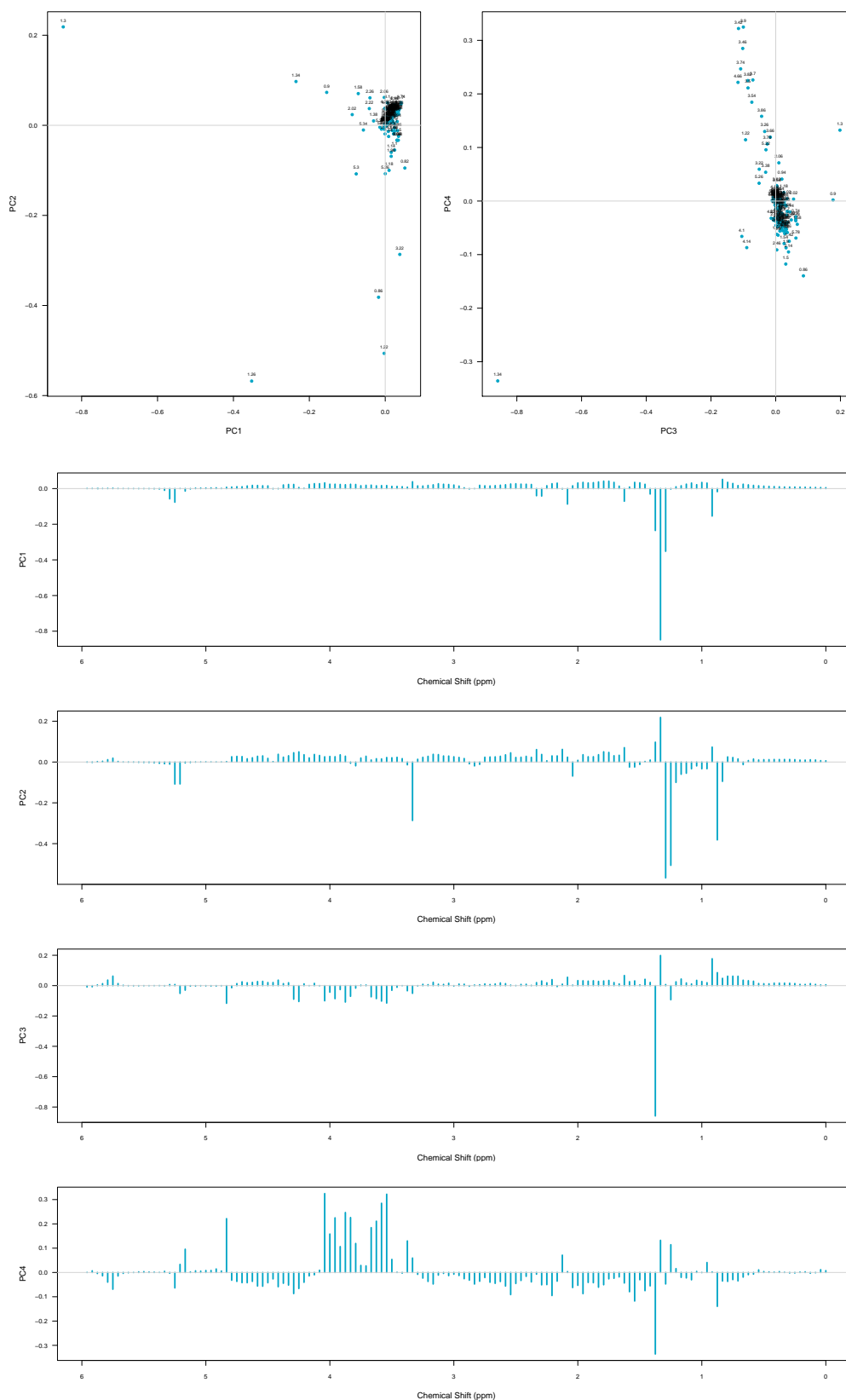
**Figure 5.10:** Scores plots for the epilepsy data superimposed with the *BMI* information. In all plots, patients in the four *BMI* categories, [16-22], (22-25], (25-28] and (28-45.1] patients are depicted in black, red, green and blue respectively. The sample numbers in all scores plots are the original ID numbers of the selected 97 patients. The data are row-scaled to a constant total.

Loadings plots can help to provide a general idea of relationships between variables, as well as between samples and variables. In general, variables that cluster most closely together are usually well correlated, so that well-correlated variables probably exhibit similar trends in the data samples. If the scores and loadings are similar in sign, then they group in samples and spectral features are correlated. A variable having very positive values in a PC in the loadings plot, is most likely to be a biomarker for a category (or group) of patients who have very high values of the scores of this PC, unlike those categories (or groups) with very low values. In other words, the extreme variables in the 1-dimensional loadings plots are likely to be biomarkers for categories of clinical characteristics with extreme values of similar sign in the scores plots for a PC.

The loadings plots for the first four PCs can be seen in Figure 5.11. In the case of the epilepsy data, there is sequential meaning to the horizontal scale of the 1-dimensional loadings plots, which relates to chemical shift in *ppm*. Low (high) scores and negative (positive) loadings, depicted as peaks in the loadings plots may mean correlation of the categories with the variables in the peaks. Variables with chemical shift in the range of 1.38–1.26 *ppm* contribute more to the variation in the first three components. Variables with chemical shifts in the range of 3.98–3.38 *ppm* contribute more to the variation of the fourth component.

Concerning *Gender*, very high negative loadings are observed for variables 1.26, 1.22, 0.86 and 3.22 on PC2. As the *Gender* categories are separated by PC2, *female* patients will tend to have larger values than *males* on these variables. On the other hand, variable 1.3 is observed to have the highest positive loading on PC2, therefore *male* patients are expected to have larger values than *females* on this variable. From Table 2 in Lindon et al. (1999), variable 0.86 is not associated with any metabolite in human blood serum, 1.22 is associated with  $\beta$ -*Hydroxybutyrate*, 1.26 is associated with *Isoleucine* (which appears also in 0.94 *ppm*), and variable 1.3 with *Fucose*. Thus, *females* may have larger intensity values in *Isoleucine* and  $\beta$ -*Hydroxybutyrate* than *males*, and the opposite may occur in the case of *Fucose*.

Regarding the *Seizure Type* of the patients, there are practically no positive loadings on PC1 (shown to be one of the PCs capable of separating the *IGE* patients), thus, it is highly improbable that any variables in PC1 could be related to *IGE* patients. However, variable 1.3 in PC2 and PC3, as well as 0.9 in PC3, are observed to have the highest positive loadings in these two PCs, thus, *IGE* patients will tend to have larger values than *LRE* patients on these variables. From Table 2 in Lindon et al. (1999), variable 0.9 is associated with  $\alpha$ -*Hydroxy-n-butyrate*, *n-Butyrate* and  $\alpha$ -*Hydroxy-n-valerate*, and variable 1.3 with *Fucose*, therefore *IGE* patients will tend to have larger intensity values in these metabolites than *LRE* patients.



**Figure 5.11:** Loadings plots of the first four PCs for the epilepsy data. The variable labels in the 2-dimensional loadings plots are the chemical shifts of the variables in the proton NMR spectrum. The data are row-scaled to a constant total.

As there is no separation of the patients with respect to their *Response to AEDs*, no categories of response are associated with high or low scores, therefore, no comparison between variables and samples can be attempted from the results of the scores and loadings for this clinical characteristic.

Concerning *Age*, there are two variables, 1.3 and 1.34, having the highest positive loadings on PC2, and the scores for the *Age* category [26-47) are high, meaning that patients in this *Age* category will tend to have larger values of these two variables than the rest of the patients. As variable 1.3 is associated with *Fucose* and variable 1.34 with the former, and *Lactate*, *Threonine* and  $\alpha$ -*Hydroxyisobutyrate* (Lindon et al., 1999), the patients in this category will tend to have larger values in these metabolites than the remaining patients in the other two *Age* categories. In addition, patients in *Age* category [47-99) will tend to have large values on variables 1.26, 1.22, 0.86 and 3.22, as both the loadings of these variables and the scores of the patients in this specific category are very low. Therefore, these patients will have larger values of the metabolites associated with these variables (seen in the previous paragraph for *Gender*). Also, patient 44, has the highest PC4 score, and considering that there is a peak of high positive loadings in the range 3.46 – 4.1 ppm of PC4, a relation between these variables and that specific patient is indicated. That is, on these variables, this patient will tend to have the highest intensity values (the variables in these chemical shifts correspond to metabolites such as *Tryptophan*, *Choline*, *Glycerol*, *Myo-inositol*, *Glycine*, *Ethanol*, *Valine*, *Isoleucine*, *Leucine*,  $\alpha$ -*Glucose*,  $\beta$ -*Glucose*, *Lysine*, *Glutamine*, *Glutamate*, *Alanine*, *Ornithine*, *Methionine*, *Betaine*, *Creatine*, *Tyrosine*, *Hippurate*, *Histidine*, *Phenylalanine* and *Creatinine*).

Things are not as clear in the case of *BMI* as for *Age*, but it can be seen in Figure 5.11 that the first two *BMI* categories, [16-22] and (22-25], despite having high PC1 scores, do not seem to be related to any variables, as practically no positive peaks exist in the loadings of PC1. Nevertheless, the variables in the range 1.26 – 1.34 ppm, as well as 0.9, may be related to the largest two *BMI* categories, with patients in these categories having larger values on these variables, and consequently on the associated metabolites mentioned previously, than the patients in the other two *BMI* categories, since both loadings and scores in PC1 have very low values.

## 5.4 Conclusions

The first part of the exploratory analysis has covered the application of PCA to the epilepsy data. The data set that was used for this purpose contained those spectral regions with chemical shifts in the range 0 – 6 ppm, and the data was row-scaled to a constant total to make the spectra more comparable. Results of the analyses indicated that the first two principal components account for approximately 95% of the total

variation in the data, therefore should be sufficient to separate the patients with respect to their clinical characteristics, *Gender*, *Seizure Type*, *Response to AEDs*, *Age* and *BMI*. Various stopping rules, including the *broken stick* and *parallel analysis*, were used to confirm that only the first two PCs should be retained. The data were examined for the possibility of existence of potential outliers and 9 patients were removed from the data, as diagnostic plots showed them to be orthogonal outliers. The analyses were re-run with the reduced data, confirming that there was no effect of these patients' exclusion from the data on the PCA results, therefore the original data was used for further analyses.

Scores plots, superimposed with the information for the five clinical characteristics, demonstrated that no PC can separate the patients with respect to their *Response to AEDs*. This was confirmed by the construction of the general linear model with explanatory variables being the first four components and dependent variable the recoded *Response to AEDs* information. Concerning *Gender*, it was shown that the first two PCs are sufficient to separate the patients, with *females* having low and *males* high scores. Similar results were observed for the *Seizure Type* of the patients, with only the first two PCs needed to separate the *IGE* from *LRE* patients. The *IGE* patients were associated with high scores, whereas the *LRE* patients were mainly associated with low scores on these two PCs. As with the *Response to AEDs* case, here also a general linear model was used to confirm the findings for the *Seizure Type*. In the case of *Age*, the scores plots illustrated that three PCs can separate the patients with regards to the three pre-defined categories of *Age*, [16-26), [26-47) and [47-99). The first *Age* category is associated with high PC1 and PC3 scores, whereas the other two categories are associated with high PC2 scores and low PC2 scores respectively. Regarding *BMI*, patients were divided into four categories, i.e. [16-22], (22-25], (25-28] and (28-45.1]. Analyses showed that the first three PCs are associated with *BMI*, but not as clearly as in the case of *Age*. PC1 can separate the patients in the first two lower *BMI* categories (high PC1 scores) from those patients in the last two higher *BMI* categories (low PC1 scores). PC2 is mainly associated with the patients in the last *BMI* category (high scores) and PC3 mainly with those patients in the third category (high scores). The first two categories cannot be separated well from each other by any of the PCs.

Loading plots were drawn to examine any relationship between the variables in the data, as well as between samples and variables. Results indicated that *female* patients seem to have larger values on variables 1.26, 1.22, 0.86 and 3.22, having large negative loadings on PC2, whereas *male* patients have larger values on variable 1.3 having the highest positive loadings on PC2. Concerning *Seizure Type*, only variable 1.3 on PC2 and PC3, and variable 0.9 on PC3 are indicative of the *IGE* patients, having larger values for them. Results for *Response to AEDs* are inconclusive, as it was shown that no PC can separate the patients with respect to this. On variables 1.3 and 1.34 (high

positive loadings on PC2), patients in the *Age* category [26-47) have larger values than the remaining patients. Variables 1.26, 1.22, 0.86 and 3.22, with negative loadings on PC2, are associated with the patients in *Age* category [47-99), who are observed to have the larger intensity values on these variables. Finally, variables in the range 3.46 – 4.1 *ppm* constitute a peak of high positive loadings on PC4, and could be related to patient 44, who has the highest PC4 score in the second *Age* category. Regarding the *BMI* categories, the variables in the range 1.26-1.34 *ppm* and 0.9 seem to be those where the patients in the last two (higher) *BMI* categories have the larger values, but there is no indication that for any variable the patients in the first two *BMI* categories have larger values than patients in the last two categories. In Chapter 9, it will be discussed which methods or statistical indicators can be used to identify which variables are most significant to discriminate between the categories of the five clinical characteristics.

In general, PCA has been quite helpful in obtaining a good idea of the general structure of the epilepsy data and the clinical characteristics of the patients, with the exception of the *Response to AEDs*, for which the technique was not capable of providing any information on whether the patients can be separated with regards to this particular clinical characteristic. In the next chapter, another unsupervised technique for data exploration and dimension reduction, multidimensional scaling (MDS), will be reviewed and applied to the epilepsy data, in order to establish if it can be proved more capable of separating the patients with respect to their *Response to AEDs* and to confirm the findings of PCA for the remaining clinical characteristics.

# Chapter 6

## Multidimensional Scaling

### 6.1 Introduction

Multidimensional scaling (MDS) covers a variety of multivariate statistical techniques in the field of multivariate data analysis. These techniques include among others *metric* and *nonmetric* MDS techniques, *Unfolding*, *Correspondence analysis* and *Individual differences scaling* (Cox and Cox, 2001). In general, MDS scaling aims to provide a representation of an observed proximity matrix by means of a mapped configuration of points in a lower dimension than the original data space. That is, MDS uses as input data the dissimilarities between all pairs of objects in a set of  $n$  objects. It attempts to represent these dissimilarities as distances between  $n$  points (corresponding to the  $n$  objects) in a lower-dimensional space (usually 2 or 3 dimensions), such that the derived distances correspond as closely as possible to the original dissimilarities (Groenen and de Velden, 2004; Williams, 2002; Izenman, 2008). The various MDS techniques differ in the way in which the correspondence of the points' distances to the objects' dissimilarities is defined.

The selection of the appropriate MDS technique totally depends on the type of the data to be analysed. More specifically, the number of "modes" and "ways" of the input data will indicate what type of analysis is the most appropriate. A *mode* in the context of MDS is each set of objects that exists in the data. For example, the dissimilarities  $\delta_{ij}$  between the epilepsy patients are one-mode data. Each index in the measurement between objects is a *way*. Thus, the dissimilarities  $\delta_{ij}$  mentioned above are two-way data, as there are two indices  $i$  and  $j$ . Correspondence and unfolding analysis usually require two-mode, two-way data, while two-mode, three-way data can be analysed using individual differences scaling, and other techniques can handle even higher-mode, higher-way data (Cox and Cox, 2001). In addition, the scale on which the dissimilarities are measured indicates whether metric or nonmetric MDS is needed. If the dissimilarities are measured on the ratio or interval scale, then metric MDS is the most appropriate, whereas if the data is ordinal or nominal (qualitative) then nonmetric

MDS is more suitable as it is concerned only with the ranks of the dissimilarities and not the actual values (Izenman, 2008; Cox and Cox, 2001).

## 6.2 Classical Scaling

*Classical scaling* algorithms are algebraic methods used for fitting  $n$   $p$ -dimensional objects into  $n$  points in a lower-dimensional space, such that the original dissimilarities  $(\delta_{ij})$  of the objects are approximated as closely as possible to the interpoint distances  $(d_{ij})$ . That is,

$$d_{ij} \approx (\delta_{ij}).$$

If  $n$   $p$ -dimensional objects are denoted by  $\mathbf{x}_i$  with  $i = 1, \dots, n$ , then a dissimilarity  $(\delta_{ij})$  between the objects  $\mathbf{x}_i$  and  $\mathbf{x}_j$  with coordinates  $\mathbf{x}_i = (x_{i1}, \dots, x_{ip})$  and  $\mathbf{x}_j = (x_{j1}, \dots, x_{jp})$  is given by

$$\delta_{ij} = \left\{ \sum_{k=1}^p |x_{ik} - x_{jk}|^q \right\}^{\frac{1}{q}} \quad (q > 0).$$

The most common  $L_p$  metric in classical MDS is the Euclidean distance (given for  $q = 2$  in the above formula). A proximity matrix  $\Delta$  is an  $(n \times n)$  matrix which contains all pairwise dissimilarities between the  $n$  objects, i.e.  $\Delta = (\delta_{ij})$ . The classical MDS algorithm can then be summarized in the following steps (Izenman, 2008; Wickelmaier, 2003; Williams, 2002; Everitt and Hothorn, 2006):

1. Given the  $(n \times n)$  proximity matrix  $\Delta = (\delta_{ij})$ , obtain the  $(n \times n)$  matrix  $\mathbf{A} = (\alpha_{ij})$  where

$$\alpha_{ij} = -\frac{1}{2}\delta_{ij}^2.$$

2. Obtain the double - centred symmetric  $(n \times n)$  matrix

$$\mathbf{B} = -\frac{1}{2}\mathbf{J}_n\mathbf{A}\mathbf{J}_n,$$

where

$$\mathbf{J}_n = \mathbf{I}_n - n^{-1}\mathbf{1}_n\mathbf{1}_n^T.$$

and  $\mathbf{I}_n$ ,  $\mathbf{1}_n$  are the  $(n \times n)$  identity matrix and the  $(n \times n)$  matrix with all elements equal to 1, respectively.

3. Compute the eigenvalues and eigenvectors of  $\mathbf{B}$ . If  $\mathbf{\Lambda} = \text{diag}(\lambda_1, \dots, \lambda_n)$  is the matrix of the eigenvalues of  $\mathbf{B}$  and  $\mathbf{V} = (v_1, \dots, v_n)$ , the matrix of  $\mathbf{B}$ 's eigenvectors arranged as columns, then by the spectral theorem,

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



4. The derived configuration is the  $t$ -dimensional configuration of the  $n$  objects given by the coordinate matrix

$$\mathbf{X} = \mathbf{V}_t \mathbf{\Lambda}_t^{\frac{1}{2}}$$

where  $\mathbf{V}_t$  is the matrix of  $t$  eigenvectors and  $\mathbf{\Lambda}_t$  is the diagonal matrix of the  $t$  largest positive eigenvalues of  $\mathbf{B}$  respectively ( $t \leq p$ ).

5. If the  $L_p$  distance metric in use is the Euclidean, then all eigenvalues of matrix  $\mathbf{B}$  are positive and the best fitting  $t$ -dimensional configuration is given by the  $t$ -largest eigenvalues. This fitting is adequate if the size of the criterion

$$P_t = \frac{\sum_{i=1}^t \lambda_i}{\sum_{i=1}^{n-1} \lambda_i}$$

which is a measure of the proportion of variation explained by using  $t$  dimensions, or *Mardia's* criterion

$$\frac{\sum_{i=1}^t |\lambda_i|}{\sum_{i=1}^n |\lambda_i|}$$

is of order of 0.8 or larger (Everitt and Hothorn, 2006). If other  $L_p$  metrics have been used, then any negative eigenvalues of matrix  $\mathbf{B}$  (with possible coordinate values being complex numbers) can either be ignored or a suitable constant  $c$  be added to the dissimilarities, e.g.

$$\delta_{ij} = \delta_{ij} + c(1 - \delta^{ij})$$

where  $\delta^{ij}$  is the Kronecker delta (Cox and Cox, 2001). The algorithm can then be executed again from the first step to obtain new coordinate values for the points in the  $t$ -dimensional space, corresponding to the  $n$  objects.

An important consideration when applying MDS techniques is the identification of the number of dimensions that the derived configuration should have, to ensure that no important information has been lost during the MDS procedure. The maximum required dimensions will be identified by examining the eigenvalues of matrix  $\mathbf{B}$ . If  $\mathbf{B}$  is positive semi-definite, as is the case when the Euclidean distance metric is used, then the number of non-zero eigenvalues is the appropriate number of dimensions, otherwise the dimensions are given by the number of positive eigenvalues. However, for practical reasons, and if the above-mentioned criteria are satisfied, it is common to use the first 2 or 3 eigenvalues, giving a reasonably small dimensional space for the derived points.

## 6.3 Metric MDS

### 6.3.1 Introduction

*Metric MDS* is applicable when the data to be analysed is measured on the ratio or interval scale. If the data contains  $n$  objects with dissimilarities  $(\delta_{ij})$ , then the requirement is to obtain a configuration such that

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

where  $d_{ij}$  are the distances between the points representing the objects in the point mapping of the original data space to the lower-dimensional space, and  $f$  is a continuous parametric monotonic function which transforms the dissimilarities into distances. Choices for  $f$  include, among others, the affine transformation ( $d_{ij} = \alpha\delta_{ij} + \beta$ ), the logarithmic transformation ( $d_{ij} = \alpha \log(\delta_{ij}) + \beta$ ), the exponential transformation ( $d_{ij} = \alpha \exp(\delta_{ij}) + \beta$ ), and the power transformation ( $d_{ij} = \delta_{ij}^\mu, \mu > 0$ ), where  $\alpha$  and  $\beta$  are unknown positive coefficients (Williams, 2002; Hebert et al., 2006).

### 6.3.2 Metric Least - Squares (LS) Scaling

*Metric LS scaling* involves the use of the least squares method to fit the distances  $d_{ij}$  to the transformation  $f(\delta_{ij})$  deriving a configuration of points such that the stress function

$$STRESS = \sum_{i < j} w_{ij} (d_{ij} - f(\delta_{ij}))^2$$

is minimized (Izenman, 2008), where  $w_{ij}$  are appropriately chosen weights. The distances  $d_{ij}$  are not restricted to be Euclidean. The choice of weights  $w_{ij}$  affects which dissimilarities will be given more weight, e.g. if  $w_{ij} = \delta_{ij}^{-\frac{1}{2}}$  then small dissimilarities between objects and the associated points are given more weight than large dissimilarities (Cox and Cox, 2001). The stress function is also considered as a goodness of fit criterion.

#### 6.3.2.1 Sammon's Non-linear Mapping (NLM)

*Sammon's non-linear mapping* is a special case of metric LS scaling, where the weighting system is

$$w_{ij} = \frac{1}{\delta_{ij}} \frac{1}{\sum_{i < j} \delta_{ij}}$$

and  $f$  is the identity function ( $f(\delta_{ij}) = \delta_{ij}$ ) (Sammon, 1969; Cox and Cox, 2001; Izenman, 2008).

The stress function in this case becomes (Sammon, 1969; Sharaf et al., 1986)

$$STRESS = \frac{1}{\sum_{i < j} \delta_{ij}^\rho} \sum_{i < j} \frac{(d_{ij} - \delta_{ij})^2}{\delta_{ij}^\rho}.$$

This method preserves the small  $\delta_{ij}$ , so that in fitting the distances  $d_{ij}$ , it gives to small  $\delta_{ij}$  greater importance than the large  $\delta_{ij}$ . This might be useful when the requirement of the analyses is to identify any clusters in the data. The exponent  $\rho$  determines whether small or large distances will prevail in weighting, e.g for  $\rho = 2$ , equal weights for small and large distances are used whereas for  $\rho = -2$  the large distances are preserved instead of the small ones (Sharaf et al., 1986). Sammon's metric stress function consists of a set of non-linear least-squares equations, which are solved using an iterative numerical procedure in order to minimize the value of the stress function (Izenman, 2008; Sammon, 1969).

## 6.4 Application of MDS to the Epilepsy Data

### 6.4.1 Introduction

The epilepsy data is of type one-mode two-way as mentioned previously. In addition, the data consists of continuous variables of quantitative nature measured on the ratio scale, as all values are non-negative due to the nature of the data (metabolite intensities). Therefore, the dissimilarities matrix of the patients contains also quantitative values and metric MDS is the most appropriate to obtain a configuration of points in a lower-dimensional space (Izenman, 2008). An initial configuration will be derived using classical scaling, which will be used as input to the NLM algorithm. The algorithm will attempt to derive a configuration as close as possible to the original, minimizing the value of the STRESS function described in the previous section. The data that will be used in the MDS analyses is the same data that was used in Chapter 5, and has been described in detail in Section 5.3.1.

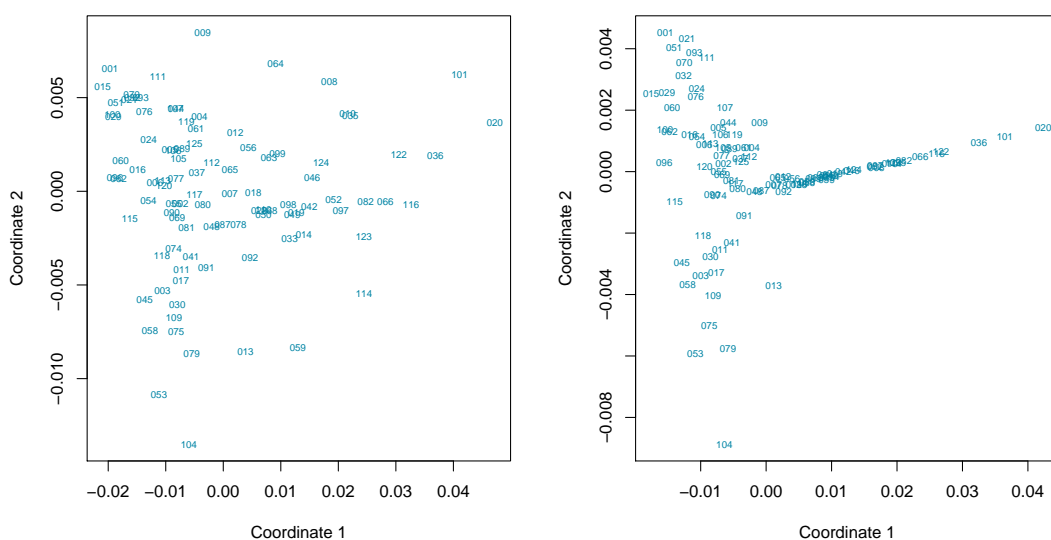
### 6.4.2 Classical Scaling Solution

Upon exploring the various distance measures that can be used to obtain an initial configuration of points, the criteria of accessing the adequacy of a 2-dimensional solution, indicated that the best distance measure in this case is the Euclidean, having a value of both criteria  $> 0.9$ , as can be seen in Table 6.1. Using a 2-dimensional solution is justified by the fact that both criteria for all metrics indicate that a very high proportion of the data variation is explained by using 2 dimensions. So, although for 3 dimensions the criteria will improve, as is reasonable, the improvement is not great (especially in

**Table 6.1:**  $P_k$  and *Mardia* criteria for various Minkowski metrics in *classical scaling* -  $k = 2$ .

Metric	$P_2$	<i>Mardia</i>
<b>Euclidean</b>	0.94949	0.94949
<b>Manhattan</b>	0.84435	0.74052
<b>Maximum</b>	0.98344	0.88297
<b>Canberra</b>	0.66185	0.52444

the case of the  $P_k$  criterion) to justify the use of a 3-dimensional space. Therefore, a 2-dimensional space should be sufficient in this case. As the *Euclidean* distance metric is the most commonly used in MDS and for both criteria its value is the same, being approximately 0.94, and suggesting that the fit is very good, it seems that it is the most appropriate to use in *classical* MDS. Results of this metric will be compared to those from the second best metric, *Maximum*. The 2-dimensional configuration derived from the classical scaling using these two distance metrics can be seen in Figure 6.1. It



**Figure 6.1:** Two-dimensional solution of *classical* MDS using the *Euclidean* (left plot) and the *Maximum* (right plot) distance metrics. The sample numbers in the plots are the original ID numbers of the selected 97 patients. The data were row-scaled to a constant total before using MDS.

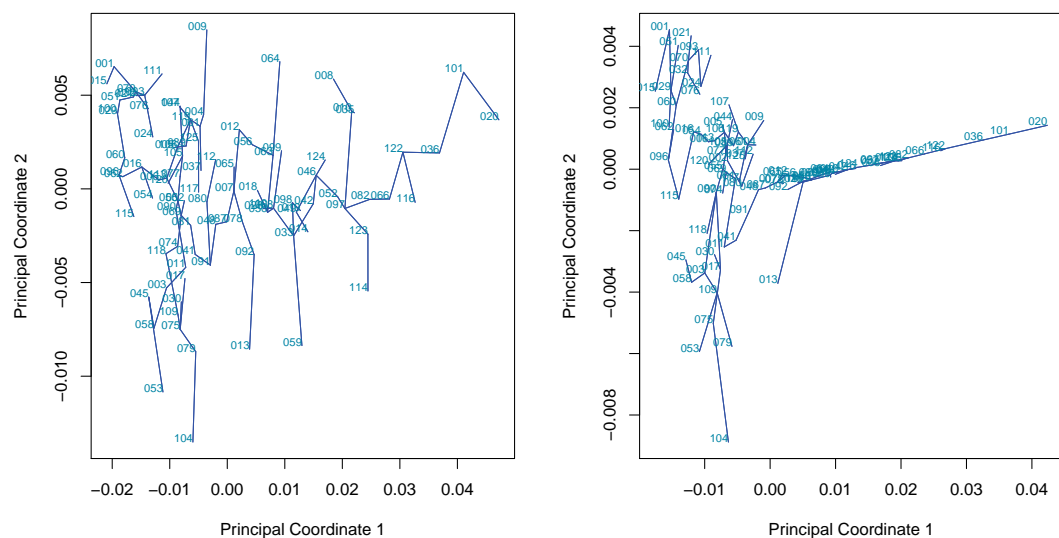
is clear that the plot for the *Euclidean* distance is similar to that of the PCA for the first two PCs in Figure 5.3, with the only difference being that the first coordinate in the MDS plot is reflected. That is expected, as in *classical* MDS using the *Euclidean* distance results to the same scores derived from PCA, except for a reflectional difference, as seen in the left panel of Figure 6.1 (Breton, 2009). The use of the *Maximum* metric results in the configuration seen in the right panel of Figure 6.1. This configuration is not affected by any rotation or reflection of the samples, in comparison to the

*Euclidean* metric plot, but succeeds in squeezing the points towards the left side of the plot with respect to coordinate 1, and towards zero in coordinate 2 for those samples with positive values in their first coordinate. This is not very helpful in identifying any groups of patients in this configuration. However, in both configurations there are no obvious groupings of the patients. Superimposing these two MDS configurations with the clinical characteristics information of the patients may show if the findings of PCA in Chapter 5, will be confirmed by MDS or even be improved.

A *spanning tree* is useful in MDS analysis, as it can provide a graphical way of highlighting any possible distortion in the MDS solution. This type of *tree* is defined as a tree spanning  $N_s$  multi-dimensional points (samples). This is any set of straight line segments joining pairs of points such that

- No closed loops occur,
- Every point is visited at least once,
- The tree has paths between any pairs of points.

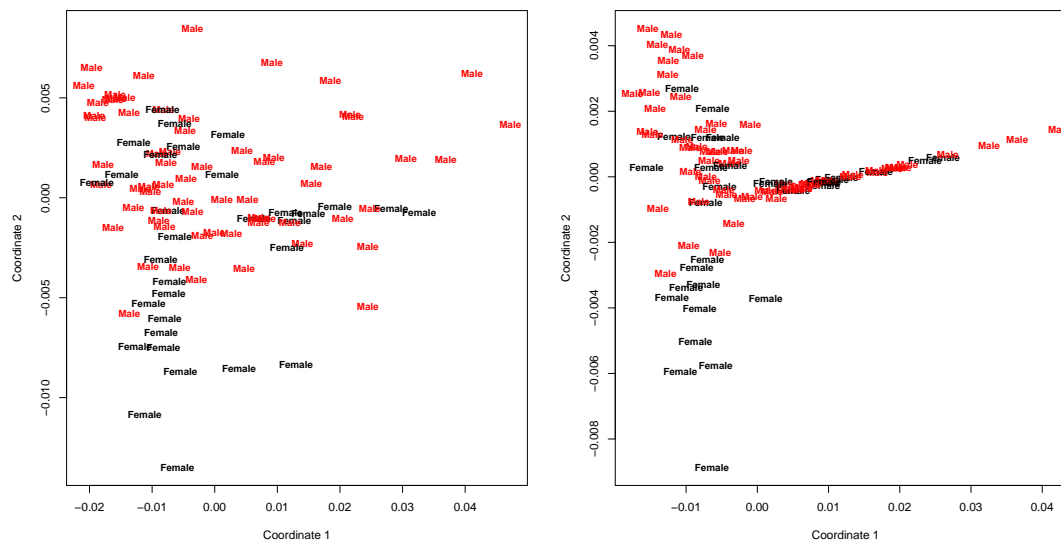
The sum of the lengths of the tree's segments is defined as the *length* of the tree. The *minimum spanning tree* (MST) is defined as the *spanning tree* with the minimum length (Everitt and Hothorn, 2006). The links of the *minimum spanning tree* can be superimposed to the 2-dimensional MDS configuration. Any distortions in the MDS solution are then identified when any nearby points on the scores plot are not connected by a direct line segment of the MST in the above MDS solution. Figure 6.2 illustrates the minimum spanning tree for the derived MDS configurations above. From the minimum



**Figure 6.2:** Minimum spanning tree for the two MDS configurations. The sample numbers in the plots are the original ID numbers of the selected 97 patients. The data were row-scaled to a constant total before using MDS.

spanning trees, it is clear that there are distortions in both models. For example, patients 13 and 59 in the *Euclidean* model, as well as 53 and 79 in the *Maximum* model, among others, appear to be quite close in the scores plot but they are not linked directly in the minimum spanning tree.

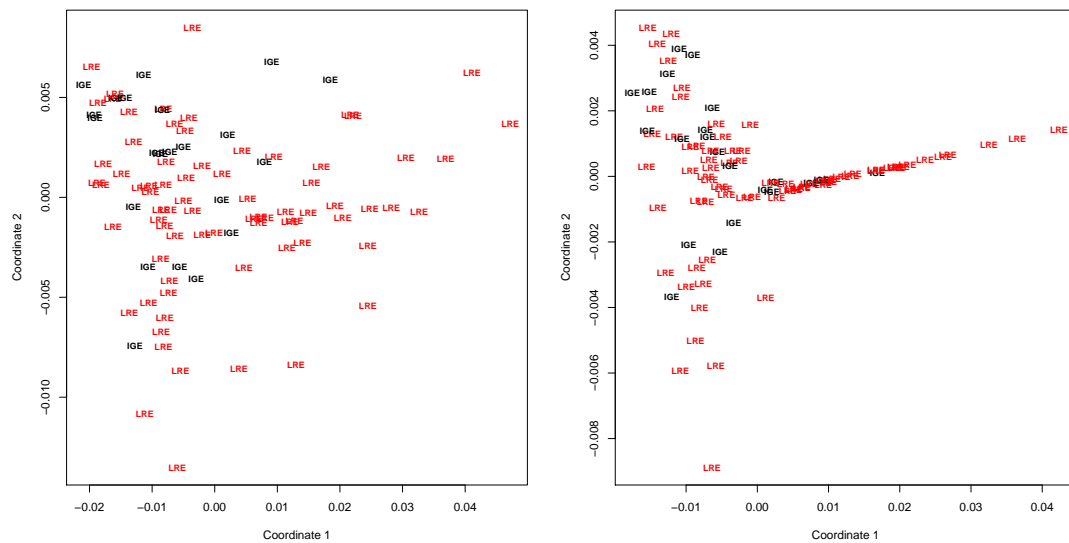
The 2-dimensional configurations derived from the classical scaling using the *Euclidean* and the *Maximum* distance measures, superimposed with the *Gender* information can be seen in Figure 6.3. Both configurations in Figure 6.3 indicate that there is



**Figure 6.3:** Two-dimensional solution of *classical* MDS using the *Euclidean* (left plot) and the *Maximum* (right plot) distance metrics, superimposed with the *Gender* information. The data were row-scaled to a constant total before using MDS.

indeed a distinction between the two categories of *Gender*, as in both cases the *male* patients are orientated towards the top of the panels and the *females* towards the bottom. This confirms the findings of PCA, which showed that PC2 separates the patients with respect to their *Gender*. Concerning *Seizure Type*, *IGE* patients are located mainly at the left side of both panels, having negative values on coordinate 1, and slightly towards the top left corner of the plots, as can be seen in Figure 6.4. However, *LRE* patients are scattered in both configurations, therefore patients are not identified as clearly as with respect to their *Gender* information. This is in agreement with the results obtained from the PCA for this clinical characteristic, as PCA and the PCR general linear model for the *Seizure Type* also showed that none of the first four PCs can separate the patients with respect to their type of seizure.

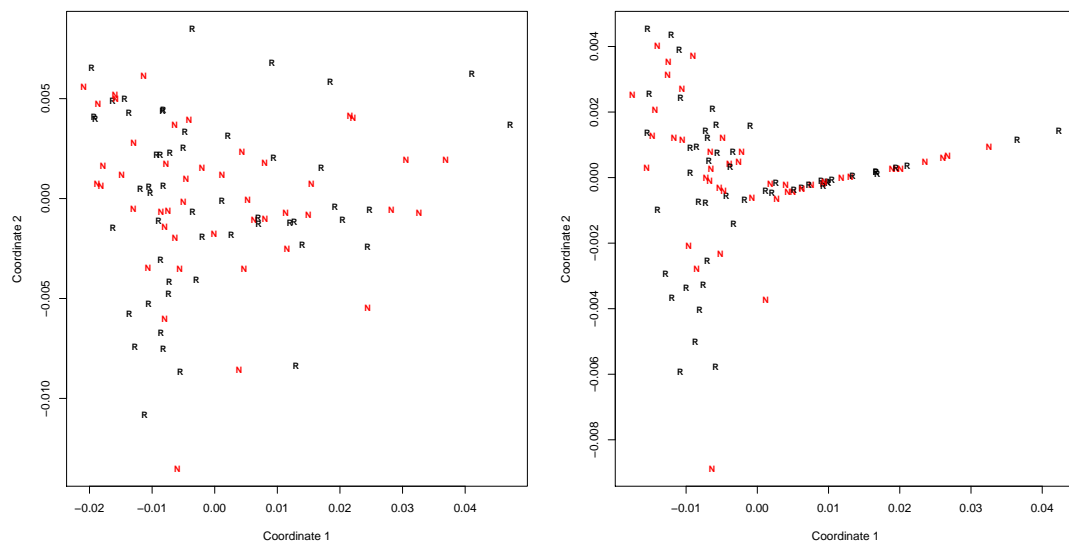
Figure 6.5 illustrates the two-dimensional MDS configurations described previously, superimposed with the *Response to AEDs* information. As it can be seen in both configurations in Figure 6.5, there are no groupings of the patients for any of the two *Response to AEDs* categories. As in the case of PCA, the two MDS models cannot



**Figure 6.4:** Two-dimensional solution of *classical MDS* using the *Euclidean* (left plot) and the *Maximum* (right plot) distance metrics, superimposed with the *Seizure Type* information. The data were row-scaled to a constant total before using MDS.

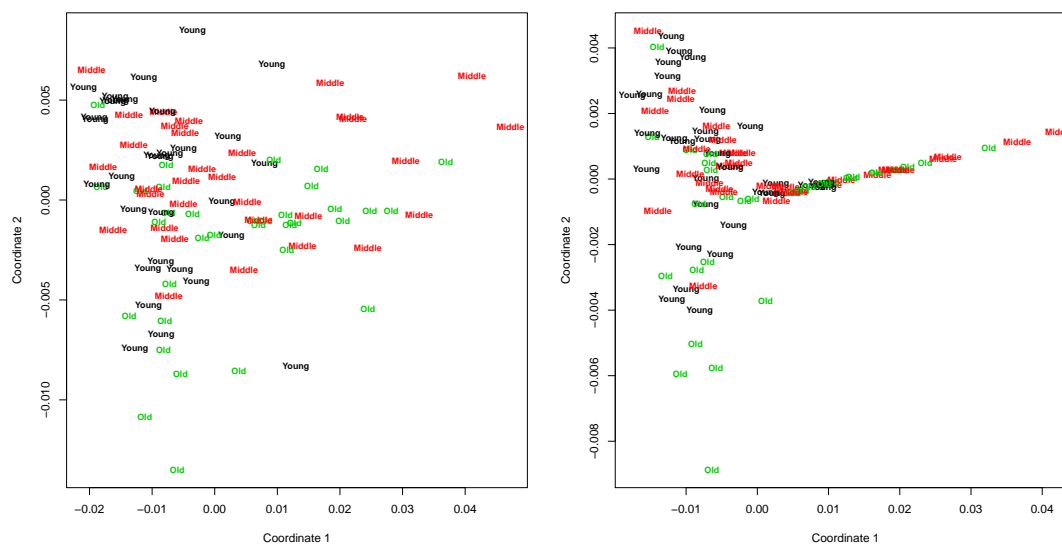
provide any helpful information on whether the patients can be separated according to their *Response to AEDs*.

Regarding *Age*, as with PCA, the same three *Age* categories, [16-26), [26-47) and [47-99), corresponding to *Young*, *Middle* and *Old* respectively, were used. The two-



**Figure 6.5:** Two-dimensional solution of *classical MDS* using the *Euclidean* (left plot) and the *Maximum* (right plot) distance metrics, superimposed with the *Response to AEDs* information. Responders and non-responders to AEDs are depicted in black (R) and red (N), respectively. The data were row-scaled to a constant total before using MDS.

dimensional MDS configurations for the two distance metrics can be seen in Figure 6.6. In this case, clustering patterns can clearly be seen, although in the *Maximum*

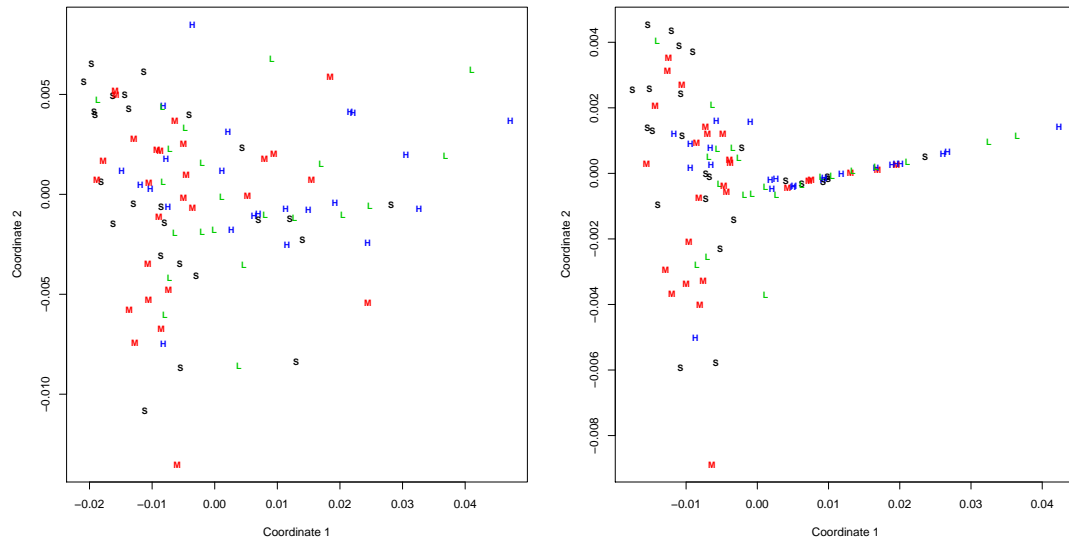


**Figure 6.6:** Two-dimensional solution of *classical* MDS using the *Euclidean* (left plot) and the *Maximum* (right plot) distance metrics, superimposed with the *Age* information. The labels of the points in the plots (*Young*, *Middle*, *Old*), correspond to the *Age* categories, [16-26), [26-47) and [47-99) respectively. The data were row-scaled to a constant total before using MDS.

configuration these are not as pronounced as in the *Euclidean* configuration, due to the compression of the points that occurs by the use of the *Maximum* distance measure. More specifically, patients belonging to the *Young* category are located to the left side of the plots in both configurations. Although in the *Euclidean* MDS model the patients in the *Middle* category are oriented towards neither the left nor right side of the plot, with the *Maximum* metric, the majority of these patients are gathered towards the left side in the configuration plot. However, patients in the two *Age* categories [26-47) and [47-99), are located towards the top and bottom of both configuration plots, respectively. The MDS findings are very consistent with the results of PCA for this clinical characteristic, as the PCA findings confirm that PC1 can separate the *Young* patients from the rest, while the patients of the other two *Age* categories are distinguishable along PC2. Therefore, the MDS models are capable of identifying the existing clustering patterns of the patients with respect to their age.

Concerning *BMI*, the four categories that are used in the analyses of the epilepsy data are [16-22], (22-25], (25-28] and (28-45.1], corresponding to *Small*, *Medium*, *Large* and *Huge* BMI values, respectively. Superimposing the *BMI* information of the patients on the two MDS configurations, the plots in Figure 6.7 are obtained. Grouping patterns are not as clear as in the case of *Age*, but some patterns can however be seen. Patients with *BMI* values in the first two *BMI* categories are located towards the left side of





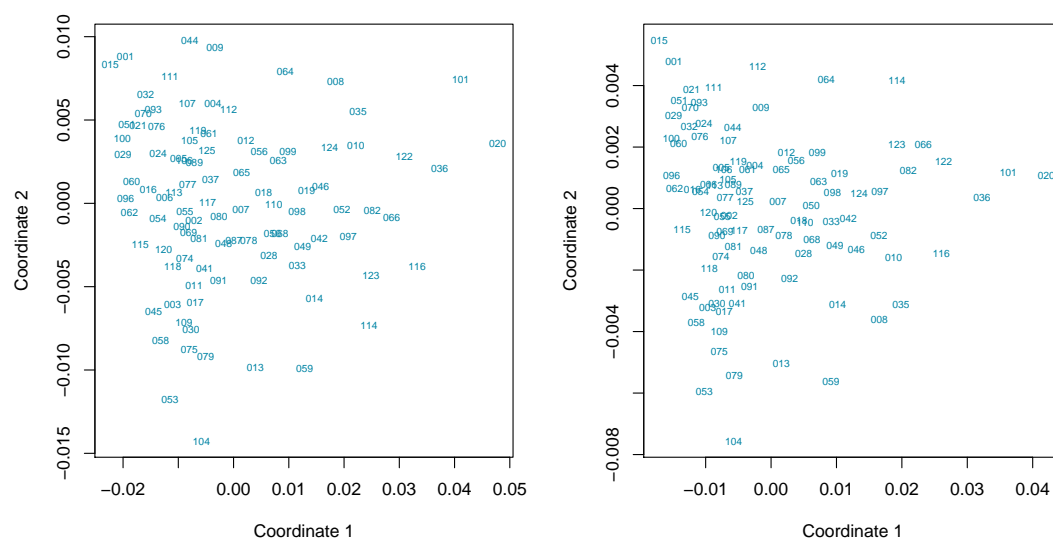
**Figure 6.7:** Two-dimensional solution of *classical* MDS using the *Euclidean* (left plot) and the *Maximum* (right plot) distance metrics, superimposed with the *BMI* information. The labels of the points in the plots (*S*, *M*, *L* and *H*) correspond to the *BMI* categories [16-22], (22-25], (25-28] and (28-45.1] respectively. The data were row-scaled to a constant total before using MDS.

the plots in both MDS configurations, whereas patients belonging to the other two *BMI* categories occupy the space on the right side of the plots. This fact is more pronounced in the *Maximum* MDS configuration, as can clearly be seen in Figure 6.7. As with PCA, it is not possible to distinguish the patients in the first two categories, as their coordinate values are in the same range for both coordinates in the plots. The latter two *BMI* categories are distinguishable along coordinate 2, as those patients with *Large* *BMI* values are located towards the centre of the plots, while the patients with *Huge* *BMI* values towards the top of the plots.

In general, *classical* MDS has been capable of confirming the results of PCA for the five clinical characteristics in question. The *Euclidean* MDS model provides a two-dimensional configuration which is easier to read than the *Maximum* MDS model, but both MDS models have proved to be useful in the pattern recognition of the epilepsy. However, so far MDS has not provided any additional information for the grouping of the patients to that obtained by PCA. In the next section, an alternative method of implementing non-linear MDS, Sammon's *non-linear mapping* will be applied to the data, to investigate if NLM can improve the results obtained from the *classical* MDS analysis.

### 6.4.3 Sammon's Non-linear Mapping (NLM) Solution

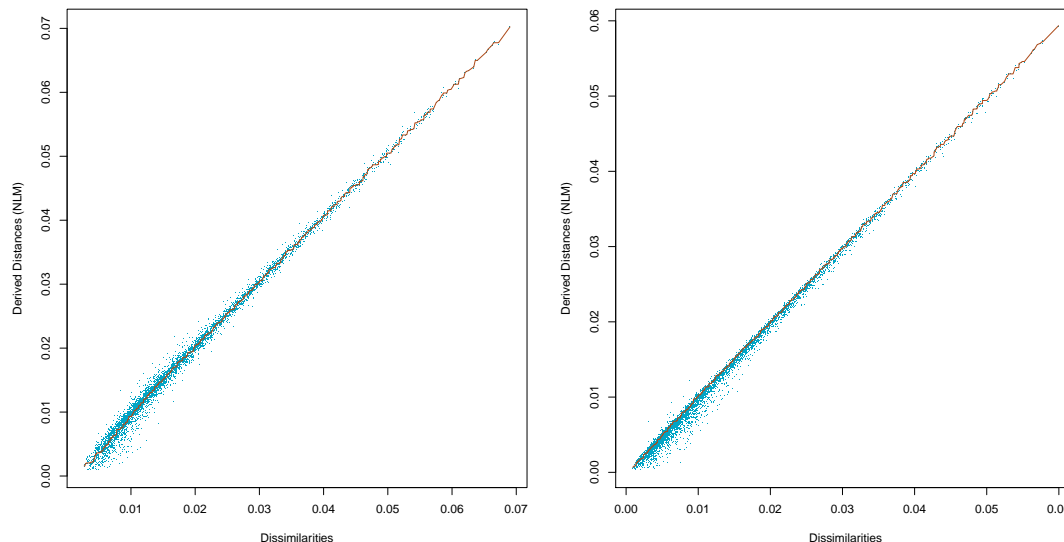
The initial configurations derived by *classical* scaling using the *Euclidean* and *Maximum* distance metrics, will be used as input to the NLM algorithm. The optimal NLM models are derived when the minimum value of the STRESS function is 0.00771 after 120 iterations and 0.00496 after 130 iterations, for the *Euclidean* and the *Maximum* NLM models, respectively. Figure 6.8 illustrates the final configurations for the two derived optimal NLM models. Comparing the NLM configurations to those obtained



**Figure 6.8:** Two-dimensional solution of *NLM MDS* using the *Euclidean* (left plot) and the *Maximum* (right plot) distance metrics. As initial configurations the *classical MDS* models depicted graphically in Figure 6.1 have been used. The sample numbers in the plots are the original ID numbers of the selected 97 patients. The data were row-scaled to a constant total before using MDS.

from *classical MDS*, it can clearly be seen that in the case of the *Euclidean* model, there is no great difference in the distances between the samples in the two MDS models and in the actual topology of the two *Euclidean* configurations, as most of the samples are located at approximately the same place in both models. On the other hand, in the case of the two *Maximum* models, the compression-like effect that occurs in the *classical MDS* model has been eliminated in the NLM model, and therefore there is a considerable difference in the between-samples distances of the formerly compressed samples. The NLM configurations are much closer in their topology than in the *classical MDS* models. In general, there is no rotation or reflection of the samples in the two NLM models, compared to the *classical MDS* configurations. Further investigation concerning the clinical characteristics of the patients might show any differences between the configurations of the two MDS methods and the two distance methods in use.

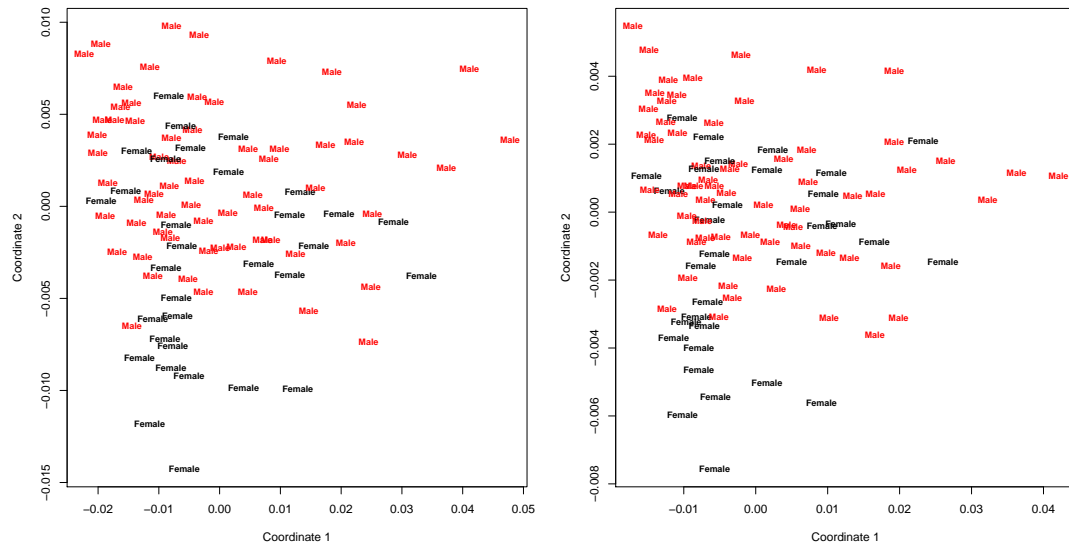
To assess the quality of the NLM solution, the differences between the distances derived by the NLM algorithm and the original distances have been plotted (Figure 6.9). Since Sammon's algorithm uses the identity function as the ratio transformation of the



**Figure 6.9:** Quality assessment of the two NLM solutions using the *Euclidean* (left plot) and the *Maximum* (right plot) distance metrics.

dissimilarities of the objects, these differences should appear as a straight line, which if extended towards zero would pass through the centre (0,0) of the plot axes. Thus, this plot shows both the transformation and the error due to the fact that the NLM solution uses only two dimensions (Groenen and de Velden, 2004). Indeed, the quality assessment of the NLM solution in Figure 6.9 indicates that there is a good approximation of the dissimilarities by the NLM solutions, although for small dissimilarities the differences tend to divert from the straight line considerably more than for large dissimilarities. The two models appear to have approximately similar goodness of fit, with only slight differences above and below the line, with the *Euclidean* model having more points above the line for dissimilarities in the range 0.005 - 0.04 than the *Maximum* model. In general, the difference points in the *Euclidean* model are more centralised to the red line of reference than those of the *Maximum* model, for which the majority of the difference points are below the red line.

The 2-dimensional configurations derived from NLM using the *Euclidean* and the *Maximum* distance measures, superimposed with the *Gender* information, can be seen in Figure 6.10. The configurations in Figure 6.10 indicate that the two categories of *Gender* are reasonably separated, with the *male* patients orientated towards the top of the panels and the *females* towards the bottom. Both NLM solutions are good, although the *Maximum* model seems to be slightly better than the *Euclidean* as it separates better the groups of patients at the top and the bottom of the plot, with the

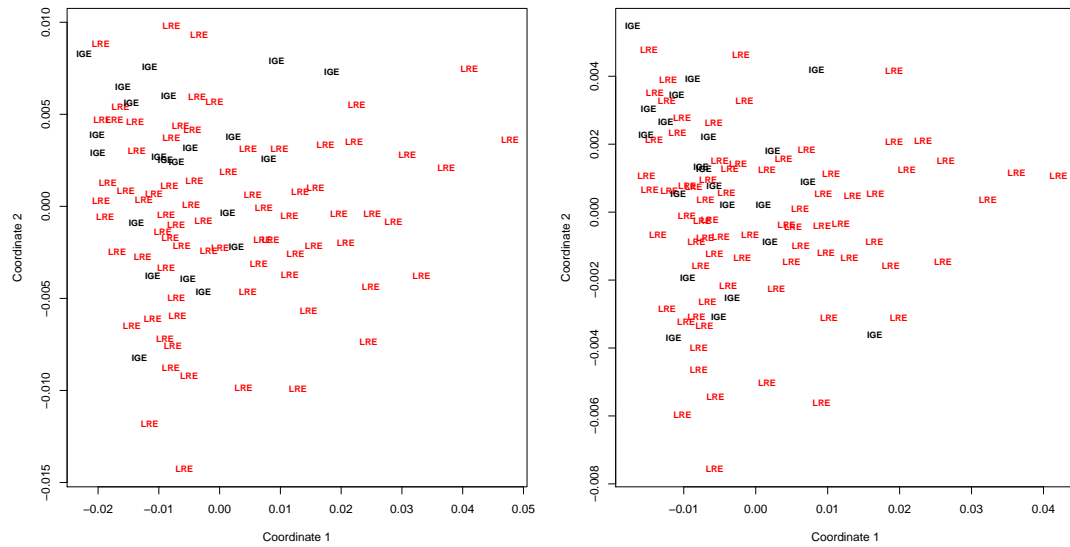


**Figure 6.10:** Two-dimensional solution of NLM using the *Euclidean* (left plot) and the *Maximum* (right plot) distance metrics, superimposed with the *Gender* information. The data are row-scaled to a constant total.

bottom group containing only *females*, while the top consists of *males* except for two *females*. However, at the right side of the plot, things look better for the *Euclidean* model, as there are no *females* at the top right of the plot, whereas in the *Maximum* configuration there is one *female*. In general, the *Euclidean* model is slightly better in the direction bottom-left to top-right, whereas the *Maximum* model shows better separation of the two categories of *Gender* strictly towards the top-bottom direction, as if this solution is slightly rotated with respect to the *Euclidean* solution.

Concerning the *Seizure Type* of the patients, *IGE* patients are located towards the top-left side of both panels in the corresponding plot, seen in Figure 6.11. As in the case of the *classical* MDS models, the *LRE* patients are located everywhere in the two-dimensional space defined by the two configurations, and they are far larger in numbers than the *IGE* patients, therefore the patients' separation with respect to their *Seizure Type* is not as easy or straightforward as in the *Gender* case.

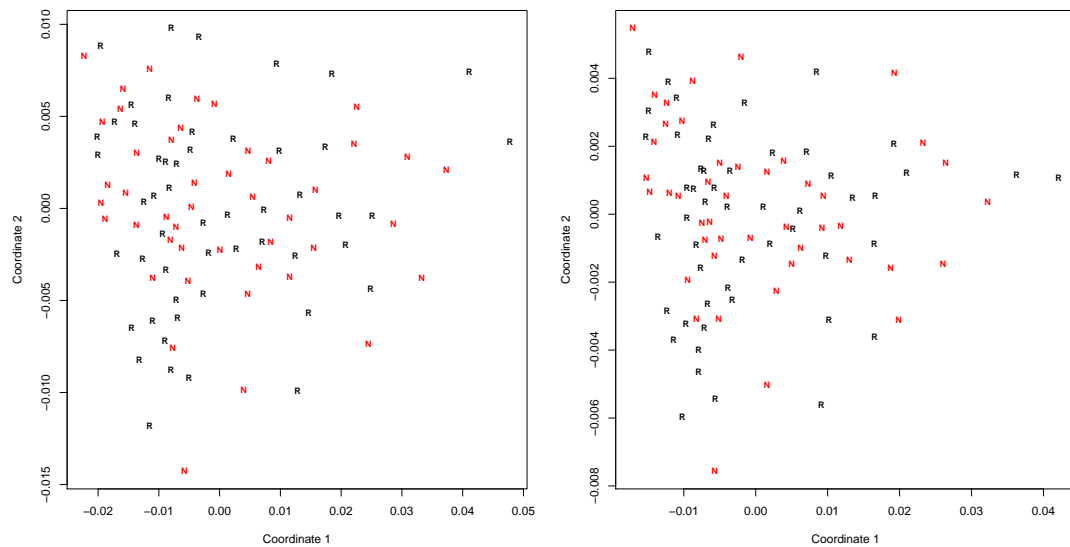
Figure 6.12 depicts the two-dimensional NLM configurations described previously, superimposed with the *Response to AEDs* information. As can be seen in both configurations in Figure 6.12, there are no grouping patterns of the patients for any of the two *Response to AEDs* categories. The *Maximum* model shows a slight inclination of the *responders* towards the left side of its corresponding configuration plot, which is not as evident in the case of the *Euclidean* model, but overall none of the two models is clearly better than the other as far as the *Response to AEDs* information is concerned. In general, NLM has not been more helpful than *classical* MDS, or PCA for that matter, in providing information on whether the patients can be separated according to their



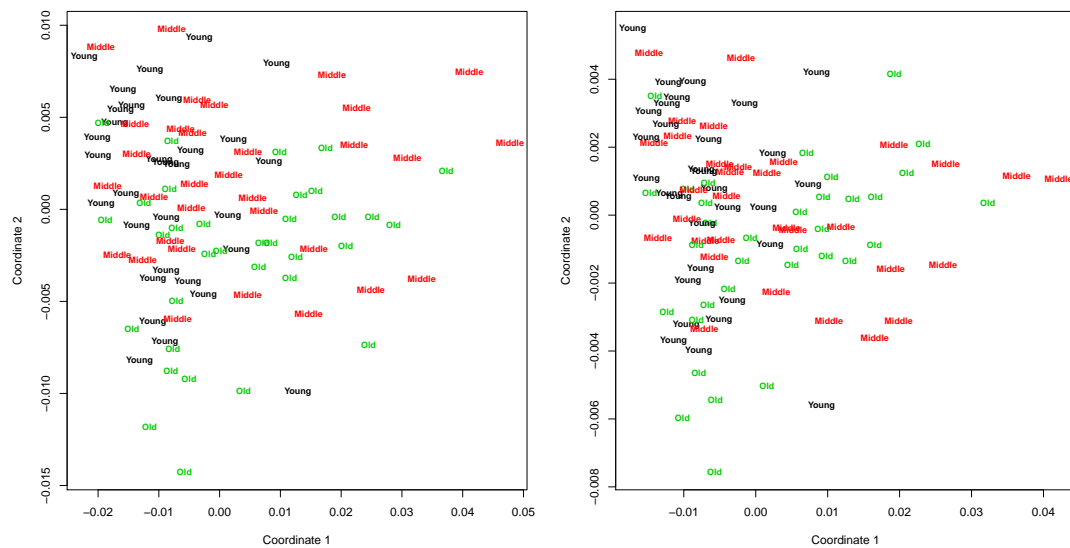
**Figure 6.11:** Two-dimensional solution of NLM using the *Euclidean* (left plot) and the *Maximum* (right plot) distance metrics, superimposed with the *Seizure Type* information. The data were row-scaled to a constant total before using MDS.

### *Response to AEDs.*

Regarding the *Age* of the patients, the two-dimensional NLM configurations for the two distance metrics with the *Age* categories superimposed can be seen in Figure 6.13. The *Maximum* NLM model is considerably better at identifying the clustering patterns



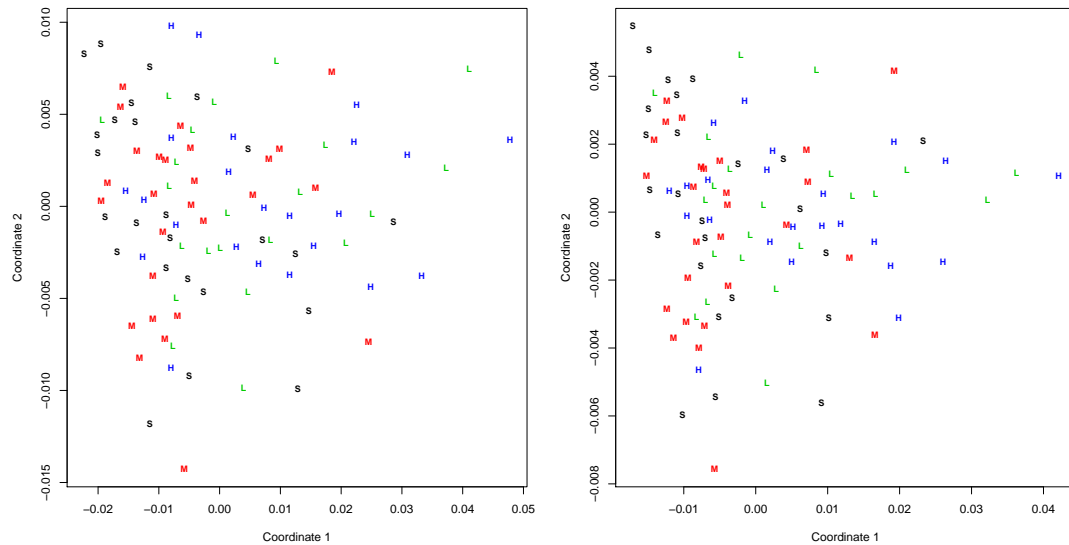
**Figure 6.12:** Two-dimensional solution of NLM using the *Euclidean* (left plot) and the *Maximum* (right plot) distance metrics, superimposed with the *Response to AEDs* information. Responders and non-responders to AEDs are depicted in black (R) and red (N), respectively. The data were row-scaled to a constant total before using MDS.



**Figure 6.13:** Two-dimensional solution of NLM using the *Euclidean* (left plot) and the *Maximum* (right plot) distance metrics, superimposed with the *Age* information. The labels of the points in the plots (*Young*, *Middle*, *Old*), correspond to the *Age* categories, [16-26), [26-47) and [47-99) respectively. The data were row-scaled to a constant total before using MDS.

for the *Age* than the *Maximum* MDS model. However, the *Euclidean* NLM model provides a clearly superior pattern recognition result to the *Maximum* NLM. More specifically, the patients belonging to the *Young* category are still located to the left side of the plots in both configurations, those patients in the *Middle* category are clearly oriented towards the top-right side of the plot in the *Euclidean* model, whereas in the *Maximum* model, most patients in this *Age* category lie along the centre and top-left part of the NLM configuration plot. Patients in the *Age* category [47-99) are located towards the bottom in the graphs of both configurations, but in the *Maximum* model some patients in this category lie towards the top-right corner of the plot. In general, NLM gives good separation results for the *Age*, as the *classical* MDS models did, with the *Euclidean* models in the two methods providing approximately similar topologies, and the *Maximum* model derived by the NLM method being far better at separating the patients in the *Age* categories corresponding to *Young* and *Middle*, than the same distance model derived by the *classical* MDS method.

Concerning *BMI*, superimposing the *BMI* information of the patients to the two MDS configurations, the plots in Figure 6.14 are obtained. Similarly to the *classical* MDS models for *BMI*, grouping patterns are not as clear as in the case of *Age*, but some patterns can however be seen. Patients with *Small* or *Medium* *BMI* values are located towards the left side of the plots in both NLM configurations, whereas patients belonging to the other two *BMI* categories lie along the right side of the plots. This is still more pronounced in the *Maximum* NLM configuration, but the differences between



**Figure 6.14:** Two-dimensional solution of NLM using the *Euclidean* (left plot) and the *Maximum* (right plot) distance metrics, superimposed with the *BMI* information. The labels of the points in the plots (*S*, *M*, *L* and *H*) correspond to the *BMI* categories [16-22], (22-25], (25-28] and (28-45.1] respectively. The data were row-scaled to a constant total before using MDS.

the two distance models are too small to be considered in any way important.

## 6.5 Conclusions

In this chapter, another data-projection method, with the advantage over PCA that it is flexible and can be used with any dissimilarity measure, is applied to the same epilepsy data as in PCA, for pattern recognition purposes, namely, *multidimensional scaling*. More specifically, two MDS methods were described in detail and used, initially the *classical* MDS, and then, the derived MDS configuration was used as input to the NLM method.

In the case of the initial configuration, results using four different distance metrics, *Euclidean*, *Manhattan*, *Maximum* and *Canberra*, were compared with the help of two criteria,  $P_2$  and *Mardia's criterion*. Considering the results of the criteria, only two metrics, the *Euclidean* and the *Maximum* were the best, giving a very good fit of the original distances of the samples to the corresponding two-dimensional MDS space. The pattern recognition capability of the two *classical* MDS models was tested both by examining the graphical representations of the models' configurations, and by superimposing the information of the available five clinical characteristics of the patients.

Results proved to be very consistent to those of PCA, but overall *classical* MDS was not capable of improving the PCA findings or adding more information to them.

Concerning the *Gender* of the patients, the two categories are distinct along coordinate 2, with the *male* and *female* patients being towards the top and the bottom of the configuration plot, respectively, in both distance models. Similarly, for the *Seizure Type*, *IGE* patients lie towards the top-left corner of the plots with both distance models being capable of showing this fact quite adequately. On the other hand, results proved that the patients do not have any grouping behaviour with respect to their *Response to AEDs*. The two *classical* MDS models are more capable of showing grouping patterns when the *Age* of the patients is considered. Clustering results are easier to see in the case of the *Euclidean* model, as the *Young* and *Middle* categories of *Age* are more separable than in the *Maximum* model. The *Young* patients lie towards the top-left corner of the plot, the *Middle* towards the right and top-right part of the plot and the *Old* towards the bottom side of the plot. Regarding the *BMI* categories of the patients, a distinction can be seen only between *Small-Medium* and *Large-Huge* BMI values, with the former pair being in the left side and the latter pair towards the right side of the plots. Once more, the separation of the patients with regards to their *BMI* values is clearly better in the *Euclidean* than the *Maximum* model, due to the compression-like effect of the points in the *Maximum* model.

In general, the *Euclidean* model proved to be slightly better than the *Maximum* in identifying any clustering patterns concerning the patients, except for the *Response to AEDs* where both distance models consistently failed to show any clustering of the patients.

Applying the NLM method to the data, using the derived *classical* MDS models as initial configuration, showed that only very slight differences are observed between the *classical* MDS and the NLM results, when the *Euclidean* distance metric is used. On the other hand, in the case of the *Maximum* models, the compression-like effect of the points that had been observed in the *classical* MDS model has been remedied in the NLM, with the *Maximum* NLM configuration being much closer to that of the *Euclidean* NLM model. This was confirmed by the quality assessment of the two NLM solutions, with the aid of Shepard-like plots (described in Section 6.4.3 and seen in Figure 6.9), which showed that both NLM models fit the original epilepsy data quite well, with the *Maximum* model fitting the data slightly better.

As far as the clinical characteristics of the patients are concerned, the results were quite close to those of the *classical* MDS models. More specifically, for *Gender*, the separation of the patients is similar to that of the *classical* MDS models, with the *Maximum* NLM model showing slightly better the differences between the two categories than the *Euclidean*. In addition, the separation in the *Maximum* model occurs towards the top-bottom direction, whereas in the *Euclidean* model it is rather better in the bottom-left to top-right of the plot. Results for *Seizure Type* were identical to those of the *classical* MDS models, with only the *IGE* patients identified as being located in



a specific place, the top-left of the plots, in both distance models. In addition, there was no improvement of the findings concerning the *Response to AEDs*, as both NLM models proved to be incapable of showing any separation of the patients regarding their *Response to AEDs*. The quality of separation of the patients in the NLM models with respect to the three *Age* categories, is quite good and very similar to that of the *classical* MDS models. Especially, the *Euclidean* model illustrates clearly that the *Young* patients lie towards the top-left corner, the *Middle* towards the top-right corner and the *Old* towards the bottom of the configuration plot. In particular, the *Maximum* NLM model shows far better separation of the patients in the three *Age* categories than the *classical* MDS model using the same distance measure. Finally, in the case of *BMI*, the findings of the two MDS methods are similar, with patients in the two smaller *BMI* categories lying at the left side of the plot and those with *BMI* in the larger two categories being towards the right part of the plot, in both distance models. The *Maximum* model provides a slightly better separation of these pairs of *BMI* categories, but no model can show any clustering separately for each *BMI* category.

The two MDS models were capable of reproducing quite successfully the findings of PCA, but they did not manage to provide any further information on the potential clustering of the patients with respect to their clinical characteristics. More importantly, in the case of the *Response to AEDs*, the MDS methods were not successful in identifying any clustering pattern among the patients, as for PCA.

The next chapter describes in detail four of the most important unsupervised classification techniques, for types of data such as metabonomics data, in the areas of *hierarchical clustering*, *partitioning methods* (fuzzy and hard clustering) and *competitive learning algorithms* (self organising maps (SOM)), in an attempt to devise suitable clustering models for the epilepsy data. These methods, as they are designed specifically to identify groupings present in the data, are expected to confirm the findings of PCA in Chapter 5 and MDS in the current chapter.

# Chapter 7

## Cluster Analysis

### 7.1 Introduction

*Cluster Analysis* includes a number of statistical techniques which aim to divide the data into groups (clusters) of samples with similar characteristics. Although these techniques belong to a wider area of statistics called *Classification*, they differ from supervised classification techniques such as discriminant analysis. That is, because they are unsupervised techniques, no samples in the data are known to belong to any derived clusters, and the number of clusters needed to identify any similarities (or dissimilarities) in the samples is not known beforehand.

Cluster analysis can be considered in two ways, either as identifying any natural groupings in the data or subdividing the data to facilitate its analysis. The latter is also known as *dissection* (Krzanowski and Marriott, 1995). Cluster analysis can be used in many applications, such as *data reduction*, *hypothesis generation* and *testing*, as well as in *group prediction* (Theodoridis and Koutroumbas, 2003). In metabonomics, the aim is to identify any natural groupings, as certain biological procedures of living organisms need to be assessed and usually certain biological reactions (e.g. drug response) to be identified. As a prediction tool, it can be used to classify new patients to already established groups of patients, according to their reaction to specific drugs. It is then possible to decide on new patients' medication, from the medication of the groups to which they have been classified.

Clustering algorithms can be divided into many categories. The most commonly used categories are the *sequential* algorithms, algorithms based on *cost function optimisation*, *genetic clustering* algorithms and *competitive learning* algorithms. Clustering techniques in some of these categories can be divided further into subcategories. For example, the hierarchical clustering algorithms can be divided, among others, into *agglomerative nesting* and *divisive* algorithms.

In this chapter, a number of clustering techniques will be described and applied to the same metabonomics data set that has been used in Chapters 5 and 6. These techniques

include *Hierarchical* methods (*Agglomerative nesting* algorithms), *Partitioning* methods (*Fuzzy* and *Hard* clustering algorithms) and *Competitive Learning* algorithms, as these are deemed in the literature (Gordon, 1981; Lindon et al., 2001; Adams, 2004) to be the most appropriate clustering techniques for types of data such as metabonomic data. Section 7.2 states the various considerations and decisions that need to be made when cluster analysis is applied. Section 7.3 covers the proximity measures which can be used to represent the data for cluster analysis. Hierarchical clustering methods are discussed in Section 7.5, with agglomerative nesting methods being described and applied to the metabonomics data. Partitioning methods can be found in Section 7.6 with the *fanny* fuzzy clustering technique and the *k*-means hard clustering algorithm described and applied to the data. Finally, *Competitive Learning* algorithms, with emphasis on SOM are described and applied to the metabonomics data in Section 7.7.

## 7.2 Clustering Considerations and Decisions

The procedure of applying a cluster analysis consists of the following considerations and decisions (Theodoridis and Koutroumbas, 2003):

- **Variable selection.** The first thing for consideration when applying a cluster analysis is to decide on the data to be used for the analysis. The selected variables must contain as much information about the required area of research as possible. The aim of the research and the reason for applying cluster analysis should direct the researcher to suitable variables for the analysis. For example, the type of variables in the data might play an important role in the decision on which proximity measure, clustering criterion and clustering algorithm to use for the analysis. In addition, pre-processing and pre-treatment of the data might be needed before their use in the analysis.
- **Proximity measures.** The selected variables, as discussed in the previous step, should contribute equally to the computation of a proximity measure, which gives an indication of how similar the paired objects in the data are. As will be seen in the next section, there are various proximity measures which can be used in cluster analysis.
- **Clustering criteria.** The type and number of clusters that are required will dictate which criterion will have to be used in the analysis.
- **Clustering algorithm.** After selecting the proximity measure and clustering criterion to be used, a specific clustering algorithm must be selected, to determine the way of obtaining the clusters from the data.
- **Cluster validation.** After performing cluster analysis, certain tests must be applied to certify the correctness of the results.

- **Cluster Interpretation.** The final step in the procedure of cluster analysis is to interpret the results of the analysis. At this step, the results from other statistical techniques, e.g. PCA, might have to be combined with those of the cluster analysis, to allow for easier interpretation.

Sometimes it is also necessary to assess the suitability of the data for cluster analysis (clustering tendency). This involves the application of various tests on the data, to establish whether the data appears to have a clustering structure or not. As is logical, the considerations and decisions mentioned above are very important in clustering, as for example, selecting different variables/descriptors, proximity measures and clustering algorithms may produce completely different clustering solutions.

## 7.3 Proximity Measures

In order to cluster a set of objects into natural groups, it is necessary to introduce the notion of proximity. As clustering involves the notion of objects being similar (or close) or dissimilar (not close) to each other in a data set, a means of expressing the closeness of objects to each other is needed. The way in which the objects in a dataset are presented for analysis plays an important role in the approach that will be used to cluster the objects into groups. There are two main ways to represent the data (Gordon, 1996):

### The Pattern (or Profile Matrix)

This is a  $(n \times p)$  matrix  $X$  with elements  $x_{ik}$ , such that  $x_{ik}$  is the observed value on the  $k^{\text{th}}$  variable for the  $i^{\text{th}}$  object ( $i = 1, \dots, n$ ,  $k = 1, \dots, p$ ). In the epilepsy data, this value is the intensity/concentration of the  $k^{\text{th}}$  metabolite observed in the  $i^{\text{th}}$  patient. Matrix  $X$  is also called the data (or input) matrix, as most clustering techniques allow the use of this data representation as input for the clustering procedure.

### Proximity Matrices

Two types of matrices are considered as proximity matrices, the *dissimilarity* and the *similarity* matrix.

- **Dissimilarity Matrix.** A dissimilarity matrix  $D$  with elements  $d_{ij}$  is an  $(n \times n)$  matrix, where  $d_{ij}$  is the dissimilarity between the  $i^{\text{th}}$  and  $j^{\text{th}}$  objects ( $i, j = 1, \dots, n$ ). A dissimilarity coefficient,  $d$ , is a function from  $\Phi \times \Phi$  to  $\mathbb{R}$ , such that

$$\begin{aligned} d_{ij} &\geq 0 & \forall i, j \in \Phi \\ d_{ii} &= 0 & \forall i \in \Phi \\ d_{ij} &= d_{ji} & \forall i, j \in \Phi \quad (\text{symmetric}) \end{aligned} \tag{7.3.1}$$

where  $i, j = 1, \dots, n$  and  $\Phi$  is the set of objects for classification (Gordon, 1981; Lukasova, 1979). In addition, if  $d$  satisfies

$$d_{ij} \leq d_{ih} + d_{hj} \quad \forall i, j, h \in \Phi \quad (\text{triangle inequality}) \quad (7.3.2)$$

then  $d$  is a distance function (Kaufman and Rousseeuw, 2005; Everitt, 1993). A measure does not need to satisfy (7.3.1) and/or (7.3.2) to be considered as a dissimilarity (Gordon, 1981; Kaufman and Rousseeuw, 2005). However, a distance measure requires both equations (7.3.1) and (7.3.2) to hold. The most well-known distance measures are the Minkowski metrics which are given by Eq. (7.3.3):

$$d_{ij}^{(q)} = \left\{ \sum_{k=1}^p w_k |x_{ik} - x_{jk}|^q \right\}^{\frac{1}{q}} \quad (q > 0). \quad (7.3.3)$$

For  $q = 1$  and  $q = 2$ , eq. (7.3.3) gives the *City block* (or *Manhattan*) and the *Euclidean* metric respectively. There are many other dissimilarity measures, either distance measures or not (Gordon, 1981; Everitt and Rabe-Hesketh, 1997; Krzanowski and Marriott, 1995; Everitt, 1993).

- **Similarity Matrix.** This is a  $(n \times n)$  matrix  $S$  with elements  $s_{ij}$ , where  $s_{ij}$  is the similarity between the  $i^{\text{th}}$  and the  $j^{\text{th}}$  objects ( $i, j = 1, \dots, n$ ). The similarity coefficient  $s_{ij}$  indicates how close (or alike) objects  $i$  and  $j$  are to each other. It takes values between 0 and 1, with 0 when objects  $i$  and  $j$  are completely dissimilar, whereas 1 means the objects have the maximal similarity. A similarity function usually satisfies the following conditions (Kaufman and Rousseeuw, 2005):

$$\begin{aligned} 0 \leq s_{ij} \leq 1 & \quad \forall i, j \in 0, \dots, n \\ s_{ii} = 1 & \quad \forall i \in 0, \dots, n \\ s_{ij} = s_{ji} & \quad \forall i, j \in 0, \dots, n \quad (\text{symmetric}). \end{aligned} \quad (7.3.4)$$

It is possible to convert similarities to dissimilarities, using an appropriate transformation (Gordon, 1981; Krzanowski and Marriott, 1995; Everitt, 1993; Kaufman and Rousseeuw, 2005).

The choice of proximity measure to use in a clustering study varies greatly. It depends on the type of the data involved in the analysis, and even more on the type of variables in the data set.

## 7.4 The Silhouette Coefficient

When applying a clustering algorithm, the derived partition needs to satisfy a number of requirements concerning the solution's quality. Questions that can be asked concerning the quality of a derived partition include the following:

- Are the 'within' cluster dissimilarities small in comparison to the 'between' cluster dissimilarities?
- Which patients are well-classified or poorly classified?
- How is the data structured?
- What is the actual number of 'natural' clusters underlying the data?

A statistic developed by [Rousseeuw \(1987\)](#) to answer these questions is the *Silhouette* coefficient. If, for any object  $i$  in the data,  $\alpha_i$  is the average dissimilarity of  $i$  to all other objects in its cluster,  $c_i$ , and  $d_{ic_j}$  the average dissimilarity of object  $i$  to all objects in cluster  $c_j$  for all clusters  $c_j$  different from  $c_i$ , then by defining

$$\beta_i = \min_{c_i \neq c_j} d_{ic_j}, \quad (7.4.1)$$

the cluster, say  $c_k$ , which satisfies Eq. (7.4.1), is called the *neighbour* of object  $i$ . That is, if object  $i$  had to be assigned to a different than its current cluster  $c_i$ , then the second best choice would be its *neighbour* cluster  $c_k$ .

The *silhouette width* for object  $i$  is then given by

$$s_i = \frac{\beta_i - \alpha_i}{\max\{\alpha_i, \beta_i\}}$$

and takes values in the range  $-1 \leq s_i \leq 1$ . The higher the  $s_i$  values towards 1 are, the more well-clustered the object  $i$  is, whereas if they are towards -1 then object  $i$  is misclassified. For values near to 0, the object  $i$  lies in the boundary between its assigned and its neighbour cluster and it is not clear any more to which of the two clusters this object actually belongs.

Similarly, the *average silhouette width*,  $\bar{s}_c$ , is the average of all  $s_i$  for cluster  $c$ . The overall average silhouette width is the average of  $\bar{s}_c$  for all objects  $i$  in the data set ([Rousseeuw, 1987](#)).

Finally, [Kaufman and Rousseeuw \(2005\)](#) define the *silhouette coefficient*,  $SC$ , as the maximum of the average silhouette widths. This can be seen as a measure of the amount of structure that has been revealed by the clustering algorithm. They also proposed the following interpretation for the values of  $SC$  (Table 7.1). An advantage of the silhouette statistic is that it depends only on the derived partition of objects and the proximity matrix, so it is not affected by the clustering algorithm that was used in the analysis of the data.

**Table 7.1:** Interpretation of the silhouette coefficient values.

<i>SC</i>	Interpretation
0.71 - 1.00	The data is very well-structured
0.51 - 0.70	The structure is reasonable
0.26 - 0.50	The structure is weak and could be artificial
$\leq 0.25$	No substantial structure has been found

## 7.5 Hierarchical Clustering Methods

### 7.5.1 Introduction

An important category of clustering techniques is the hierarchical clustering algorithms. These algorithms require more than one step to complete the procedure of establishing clusters in a data set. In addition, these techniques belong to the *hard* or *crisp* clustering methods, as each object in the data set is assigned to exactly one cluster in each partition (step) of the algorithm.

*Hierarchical cluster analysis* (HCA) has been used extensively in many studies such as the identification and classification of micro-organisms, e.g. clinical isolates of the bacterion *Salmonella enteridis* (Seltmann et al., 1994), and the development of animal or other models for toxicological studies of drug candidates, e.g. the discrimination between control rat populations and rats subjected to treatment with bacterial lipopolysaccharide, co-administered with ranitidine to induce hepatotoxicity in rats, in an attempt to develop a predictive model of idiosyncratic toxicity (Harrigan et al., 2004).

Examples of the use of HCA in metabolite profiling are the identification of similarities in the metabolite features observed between a number of different serum extraction methods applied to LC-MS generated metabolic profiles of human serum (Want et al., 2006) and the comparison and clustering of metabolic profiles of the kidney and urine of three wild small mammals and the laboratory rat, obtained by  $^1\text{H}$  NMR spectroscopy (Griffin et al., 2000).

Two important types of HCA algorithms are the *agglomerative nesting* and the *divisive* algorithms. A detailed description of the former is given in Section 7.5.2, while a brief mention to the latter can be found in Section 7.5.3. The application of HCA to the epilepsy metabonomics data is applied in Section 7.5.4.

### 7.5.2 Agglomerative Nesting Algorithms

In these algorithms, the analysis starts with  $n$  clusters containing only one sample each, and ends at one cluster containing all samples in the data set (Everitt, 1993). In general, at each step of the procedure the agglomerative algorithm finds the nearest (with regards to a pre-specified dissimilarity criterion) pair of different clusters, merges

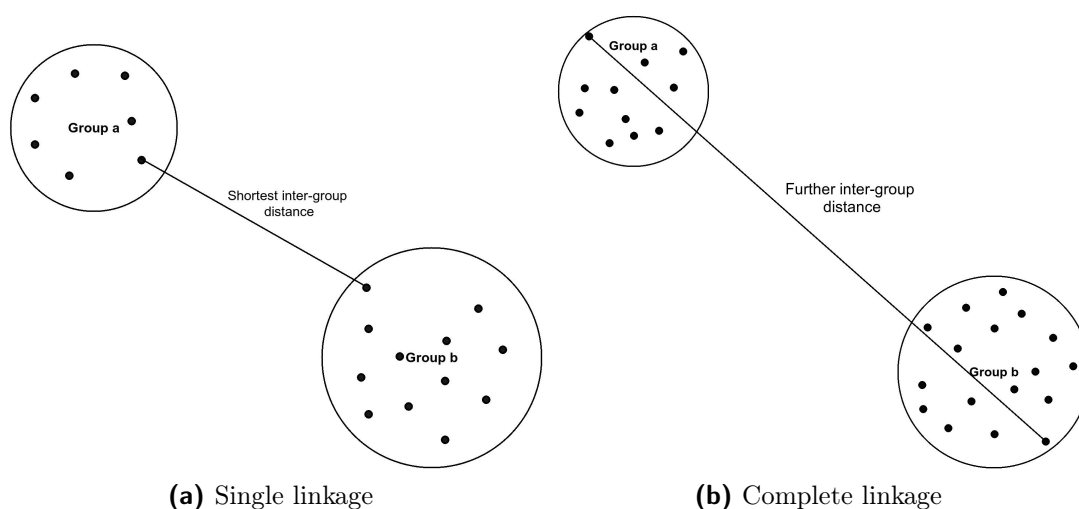
them and decreases the number of clusters by one. When the number of obtained clusters equals one (all samples have been merged into one cluster), then the procedure ends. The outcome of these techniques depends on the chosen distance or similarity measure for the clusters.

Agglomerative nesting techniques are used almost exclusively in HCA studies, especially in studies involving *Fourier Transform Infrared Spectroscopy* (FTIR) generated data (Mariey et al., 2001). They are also quite popular in NMR metabonomics studies such as the identification of similarities in metabolic profiles generated by  $^1\text{H}$  NMR spectroscopy of urine samples collected from control and dosed (with 19 model compounds in different doses) rats, investigating the metabolic effects and toxicity at different pre-specified time points (Beckonert et al., 2003).

There are many agglomerative techniques, with the most commonly used being the following:

### Single Linkage

Known also as the *Nearest Neighbour* algorithm, this is one of the simplest methods. The distance between groups is given by the closest (or furthest if a similarity measure is used) pair of samples, where each pair consists strictly of one sample from each group (Figure 7.1). In this algorithm, the dissimilarity between two clusters is defined as in



**Figure 7.1:** Illustrated example of single and complete linkage methods

the following equation

$$d_{C_p C_q} = \min_{\substack{i \in C_p \\ j \in C_q}} d_{ij},$$

where  $C_p$  and  $C_q$  are any two clusters and  $i, j$  are samples in clusters  $C_p$  and  $C_q$  respectively. In general, the clusters obtained by single linkage are formed at low dissimilarities



in the dissimilarity dendrogram (tree diagram such that in Figure 7.11). Therefore, this algorithm is especially suitable for identifying elongated clusters.

### Complete Linkage

This method is also known as the *farthest neighbour* algorithm. It is the opposite of single linkage, due to the fact that the distance between groups is now that of the farthest distant pair of samples, one sample from each group (Figure 7.1). The dissimilarity between two clusters in this case is given by the equation below

$$d_{C_p C_q} = \max_{\substack{i \in C_p \\ j \in C_q}} d_{ij},$$

where  $C_p$ ,  $C_q$ ,  $i$  and  $j$  are defined as in the single linkage case. Contrary to single linkage, in complete linkage the obtained clusters are formed at high dissimilarities in the dissimilarity dendrogram, As the complete linkage is more capable of identifying small, compact, spherical clusters, it should be the preferred method if there is any evidence that compact clusters exist in the data.

### Unweighted Pair-group Method Average (UPGMA)

This method is also called *Average* linkage. In this case, the distance between two clusters is given by the average of all the dissimilarities between the samples of one cluster and the samples of the other cluster. Each pair contains one sample from each cluster. The distance between two clusters is given by Equation (7.5.1)

$$d_{C_f C_q} = \frac{n_i}{n_i + n_j} d_{C_i C_q} + \frac{n_j}{n_i + n_j} d_{C_j C_q} \quad (7.5.1)$$

where  $C_f$  and  $C_q$  are the newly formed and the old clusters respectively, and  $n_i$ ,  $n_j$  the cardinalities of clusters  $C_i$  and  $C_j$  respectively (Theodoridis and Koutroumbas, 2003). This method is a compromise between the two extreme methods, single and complete linkage, producing relatively spherical clusters (Kaufman and Rousseeuw, 2005).

### Weighted Pair-group Method Average (WPGMA)

This method, also called the *McQuitty* method, is a variant of the group average method, described previously, whose dissimilarity between two clusters is defined as Equation (7.5.2)

$$d_{C_f C_q} = \frac{1}{2} d_{C_i C_q} + \frac{1}{2} d_{C_j C_q} \quad (7.5.2)$$

where  $C_f$ ,  $C_q$ ,  $n_i$  and  $n_j$  are as in the unweighted average case (Theodoridis and Koutroumbas, 2003). Similarly to UPGMA, this method is a compromise between the two extreme methods, single and complete linkage, producing relatively spherical clusters (Kaufman and Rousseeuw, 2005).

### Centroid Method

This method is also called the *Unweighted pair-group method using Centroids* (UP-GMC). In this case, groups are represented by their mean vectors for each variable, so that the distance between two groups, is now the distance between their two mean vectors. The distance is given by Equation (7.5.3)

$$d_{C_p C_q} = \|\bar{x}_p - \bar{x}_q\| \quad (7.5.3)$$

where  $\bar{x}_p$  and  $\bar{x}_q$  are the centroids of clusters  $C_p$  and  $C_q$  respectively. This distance is the Euclidean distance between the centroids of the clusters. The centroid method is affected by the clusters' sizes, since if the two clusters to be paired are of very different size, then the centroid of the newly created cluster tends to be too close to that of the larger of the old ones. The *Median* method (also called the *Weighted pair-group method using centroids* (WPGMC)), can overcome this problem by assuming that both clusters to be paired are of equal size, ensuring that the centroid of the newly formed cluster is always between those of the two old clusters (Kaufman and Rousseeuw, 2005).

### Ward's Method

Each time that two clusters are fused to form a new one, loss of information is certain (Everitt, 1993; Krzanowski and Marriott, 1995). Ward's method attempts to minimise that loss, by introducing a measure of the tightness of a cluster. In this case, the distance between two clusters is defined as the sum of squared Euclidean distances between the objects  $x_i$  of the cluster and its centroid  $\bar{x}_C$  (Equation (7.5.4))

$$ESS(C) = \sum_{i \in C} \|x_i - \bar{x}_C\|^2. \quad (7.5.4)$$

Other agglomerative hierarchical algorithms are *Gower's Method* and the *Flexible Strategy* (Kaufman and Rousseeuw, 2005; Gordon, 1996). Lance and Williams (1967) developed a general recurrence formula,

$$d_{(C_i \cup C_j) C_k} = \alpha_i d_{C_i C_k} + \alpha_j d_{C_j C_k} + \beta d_{C_i C_j} + \gamma |d_{C_i C_k} - d_{C_j C_k}| \quad (7.5.5)$$

to allow the evaluation of the dissimilarity between fused clusters  $C_i \cup C_j$  and another cluster  $C_k$ . Equation (7.5.5) allows the use of any clustering algorithm by selecting suitable values for the coefficients in the equation. As all the above mentioned algorithms are represented graphically by means of a plot called a *dendrogram*, Jambu (1978), introduced to this general Equation (7.5.5) the notion of the height,  $h$  of a cluster in such a plot. The updated Equation for the evaluation of the distance between fused clusters  $C_i \cup C_j$  and another cluster  $C_k$  can be seen in Equation (7.5.6).

$$\begin{aligned}
d_{(C_i \cup C_j)C_k} &= \alpha_i d_{C_i C_k} + \alpha_j d_{C_j C_k} + \beta d_{C_i C_j} + \\
&\quad + \gamma |d_{C_i C_k} - d_{C_j C_k}| + \\
&\quad + \delta_i h_{C_i} + \delta_j h_{C_j} + \epsilon h_{C_k}
\end{aligned} \tag{7.5.6}$$

where  $h_{C_i}$  is the height of cluster  $C_i$  in the dendrogram. The height (or similarity value) of a node in a dendrogram, is proportional to the distance between corresponding clusters (Ebbels, 2007). That is, clusters similar to each other are merged at low heights, whereas clusters with higher dissimilarity are merged higher up in the dendrogram. The greater the distance between heights at which clusters are merged, the easier the identification of any structure in the data is (Izenman, 2008). One can obtain a specified number of clusters (called a partition of a dendrogram) by breaking the dendrogram at an appropriate similarity level. That is, if for example, a vertically-drawn dendrogram is cut by a horizontal line at a given height, then the obtained partition contains a number of clusters equal to the number of vertical lines cut by the horizontal line. Each such intersection of a horizontal to a vertical line represents a cluster with its contents (members) being all items lying at the end of all branches below the intersection.

A list of the most well-known and used agglomerative clustering strategies and the values for the coefficients  $\alpha_i$ ,  $\beta$ ,  $\gamma$ ,  $\delta_i$  and  $\epsilon$ , to obtain the algorithms from Jambu's general equation can be seen in Table 7.2 (Gordon, 1996).

The algorithm for agglomerative nesting clustering can be summarised in the following steps (Izenman, 2008):

1. *Input:* A set of multivariate samples,  $\Omega = \{x_i, i = 1, 2, \dots, N_c\}$ , where  $N_c$  is the number of clusters, with each cluster being a singleton.
2. Compute the  $(N_c \times N_c)$  dissimilarity matrix  $D = (d_{ij})$  between the  $N_c$  clusters, where  $d_{ij} = d(x_i, x_j)$ ,  $i, j = 1, 2, \dots, N_c$  and  $d$  is a pre-selected dissimilarity measure.
3. Find the smallest dissimilarity, say  $d_{C_i C_j}$ , in the dissimilarity matrix and merge clusters  $C_i$  and  $C_j$  to form a new cluster  $C_{ij}$ .
4. Compute the dissimilarities,  $d_{C_{ij} C_k}$ , between the newly formed cluster  $C_{ij}$  and all other clusters  $C_k \neq C_i, C_j$ , using a pre-selected agglomerative method.
5. Create a new  $((N_c - 1) \times (N_c - 1))$  dissimilarity matrix, say  $D^{(2)}$ , removing from matrix  $D$  rows and columns  $C_i$  and  $C_j$  and adding a new row and column  $C_{ij}$ , using the computed dissimilarities in step 4.
6. Repeat steps 3, 4, and 5 for  $N_c - 1$  times. Hence, at the  $i^{th}$  step, the dissimilarity matrix  $D^{(i)}$  is a symmetric  $((N_c - i + 1) \times (N_c - i + 1))$  matrix,  $i = 1, 2, \dots, N_c$ . At the last step ( $i = N_c$ ),  $D^{(N_c)} = 0$ , as all clusters have been merged into a single cluster.

**Table 7.2:** Clustering techniques obtainable from the general formula of Jambu (1978).

Strategy	$\alpha_i$	$\beta$	$\gamma$	$\delta_i$	$\epsilon$
Single linkage	$\frac{1}{2}$	0	$-\frac{1}{2}$	0	0
Complete linkage	$\frac{1}{2}$	0	$\frac{1}{2}$	0	0
Average	$\frac{n_i}{n_i + n_j}$	0	0	0	0
McQuitty	$\frac{1}{2}$	0	0	0	0
Centroid	$\frac{n_i}{n_i + n_j}$	$\frac{-n_i n_j}{(n_i + n_j)^2}$	0	0	0
Sum of squares	$\frac{n_i + n_j}{n_i + n_j + n_k}$	$\frac{n_i + n_j}{n_i + n_j + n_k}$	0	$\frac{-n_i}{n_i + n_j + n_k}$	$\frac{-n_k}{n_i + n_j + n_k}$
Ward's method	$\frac{n_i + n_j}{n_i + n_j + n_k}$	$\frac{-n_k}{n_i + n_j + n_k}$	0	0	0
Median	$\frac{1}{2}$	$-\frac{1}{4}$	0	0	0
Flexible	$\frac{1}{2}(1 - \beta)$	$\beta$	0	0	0

Note:  $n_i$  is the number of objects in cluster  $C_i$ .

7. *Output:* A list of which clusters are merged at each step, the dissimilarity value (*height*) of each merge, and a summary of the procedure in the form of a dissimilarity dendrogram.

### 7.5.3 Divisive Clustering Algorithms

In this case, the clustering procedure starts with one cluster containing all samples in the data and proceeds at each step of the algorithm to increase the number of clusters by one until there are  $n$  clusters with one sample in each of them. If there are  $n$  samples in the data, there are  $2^{n-1} - 1$  non-trivial ways of dividing the samples into two clusters, therefore it is computationally infeasible to examine all possible divisions, even for cases with a moderate number of samples. As these algorithms require far more calculations than agglomerative methods do, they are not so popular and these will not be described in detail or used in the application of HCA to the epilepsy metabonomics data.

## 7.5.4 Application of HCA to the Epilepsy Data

### 7.5.4.1 Introduction

The data that will be used in the hierarchical clustering analyses is the same as in the analyses of Chapters 5 and 6. That is, the data set contains the selected 97 patients with the 144 variables in the proton NMR chemical shift range 5.98 – 0.02 ppm. The data is row-scaled to constant total.

The distance matrix of the samples will be computed using four different distance measures, *Euclidean*, *Manhattan*, *Maximum* and *Canberra*, for comparison purposes. Seven different agglomerative nesting methods will be used in order to perform the HCA. These include *Single linkage*, *Complete linkage*, *Average linkage*, the *McQuitty* method, the *Centroid* method, the *Median* method and *Ward's* method.

To facilitate identification of the best clustering method for the epilepsy data among the 28 mentioned in the previous paragraph, various statistics will be computed and plotting tools will be used to compare the results of the clustering analyses. These tools include *banner* plots, the *agglomerative coefficient*, the *cophenetic* correlation, the *Gower* distance and the *silhouette* coefficient and plot. In addition, the optimal number of clusters will be identified with the help of plotting tools such as graphs of *fusion levels* and *silhouette widths*. An important consideration in these analyses is that, although the above mentioned tools may prove sufficient to show the best hierarchical clustering method with regards to the available data, it is not necessary that a clustering between responders and non-responders to AEDs will be shown in the selected clustering method.

Three non-parametric statistical tests will be used to assess whether:

1. The clusters in the derived clustering partitions are homogeneous with respect to the proportion of observations in each of the categories which the clinical characteristics are divided into, and/or
2. The medians of the populations represented by the derived clusters are equal.

The first assessment will be tested with the  $\chi^2$  test for homogeneity and Fisher's exact test (called  $\chi^2$  and Fisher's test respectively for brevity). The  $\chi^2$  test is used when the data consists of two or more independent samples (in this case the derived clusters in each partition) categorized on a single dimension of a number of categories (in this case the categories of an appropriate clinical characteristic) (Sheskin, 2000). Hence, these tests will be used with the contingency tables for the clinical characteristics *Gender*, *Seizure Type* and *Response to AEDs*, as their values are nominal. The reason for using Fisher's test in addition to the  $\chi^2$  test is the requirement of  $\chi^2$  for the expected frequencies of all cells in the table to be of value 5 or greater (one commonly used criterion of a sufficiently large sample size for the test to be valid). This is not a requirement for Fisher's test. Therefore, whenever an expected frequency in any cell of a contingency table is less than 5, Fisher's test will be used to assess the homogeneity

of the clusters.

For the second assessment, the *Kruskal-Wallis rank sum test* (referred to as the *KW* test for brevity) will be used. This test involves ordinal (rank-order) data in a design with two or more independent samples (in this case the derived clusters). The raw values of the clinical characteristics *Age* and *BMI* will be tested using this test. As these values are in a ratio format, they will be transformed into a rank-order format to conform with the test's requirements.

#### 7.5.4.2 Comparison of Hierarchical Clustering Results

A *banner* can be considered as a horizontal barplot (see Figure 7.2) depicting graphically the agglomerative (or divisive) clustering. The values on the *x*-axis of the plot are the heights (levels) at which a merge of observations or clusters occurs, scaled from 0 for the very first merge, to the level value of the very last (final) merge. The level values can be scaled to 0-1 by dividing each level value by the maximum level value. The overall width of a banner is important as it gives an idea of the amount of structure that has been found by the algorithm. When the between-cluster dissimilarities (and consequently the highest level) are much larger than the within-cluster dissimilarities, there is a clear cluster structure in the data, and the widths of the bars in the banner are longer. The *agglomerative coefficient* can be calculated from such a plot, by taking the average of all the widths of the bars in the 0-1 scaled banner (Kaufman and Rousseeuw, 2005). The labels on the *y*-axis (right side of the bars) correspond to a permutation of the original observations, such that the creation of a dendrogram with this ordering and merge information does not have any crossings of the branches. A *banner* will be plotted using the R function `bannerplot()` of package `cluster`.

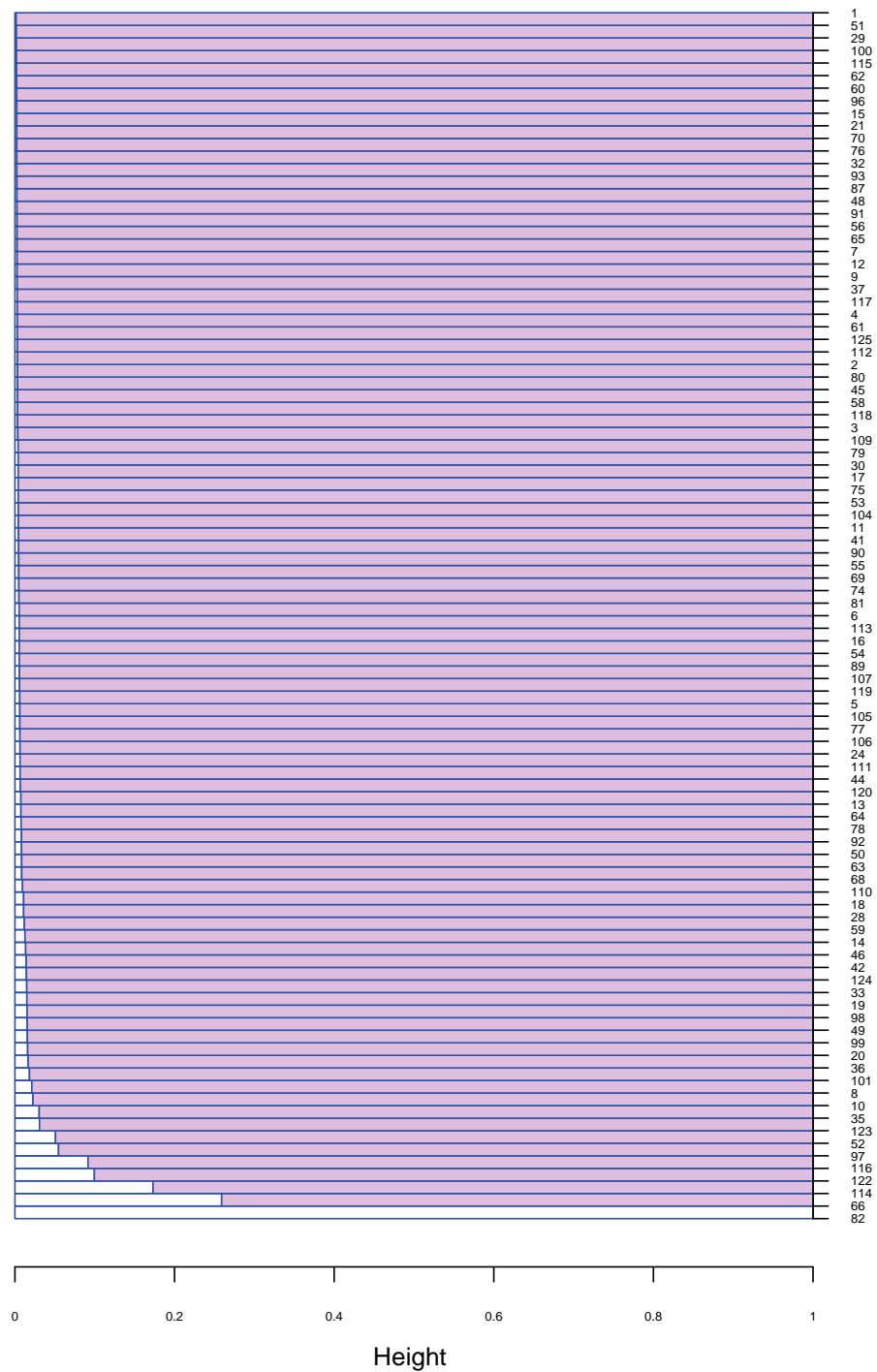
The *agglomerative coefficient* (AC), will be used to assess whether HCA finds natural structure in the data or not. The AC will be calculated using R function `calculateAC()`, developed for this purpose. This coefficient is a dimensionless quantity with values between 0 and 1. If the AC for a specific agglomerative analysis is small, then no clusters exist in the data, hence the data consist of one big cluster. The closer to 1 the value of AC is, the clearer the clustering structure of the data is (the better the agglomerative method worked to identify clusters). However, the AC value can be affected by the existence of outliers in the data, so that it is necessary when AC is large to examine also the graphical output of the clustering analysis, such as *dendrograms* and *silhouette plots*, to ensure that the value of AC is representative of the clustering structure of the data. Table 7.3 gives the agglomerative coefficient values obtained from the analyses of the 28 hierarchical clustering methods described previously. From Table 7.3, it is clear that the agglomerative coefficient has the highest value, 0.995, for the method obtained by the *Maximum* distance metric using *Ward's* agglomerative method. In general, the *Maximum* metric seems to give the best results for all the available agglomerative

**Table 7.3:** Agglomerative coefficients for the epilepsy (ROW-SCALED) data. In bold is shown the clustering method with the largest agglomerative coefficient.

Metric	Agglomerative Method						
	Single	Complete	Average	Ward	McQuitty	Median	Centroid
<b>Euclidean</b>	0.449	0.926	0.834	0.991	0.867	0.882	0.874
<b>Manhattan</b>	0.386	0.879	0.768	0.986	0.817	0.829	0.750
<b>Maximum</b>	0.644	0.957	0.936	<b>0.995</b>	0.934	0.937	0.926
<b>Canberra</b>	0.537	0.839	0.668	0.973	0.760	0.810	0.718

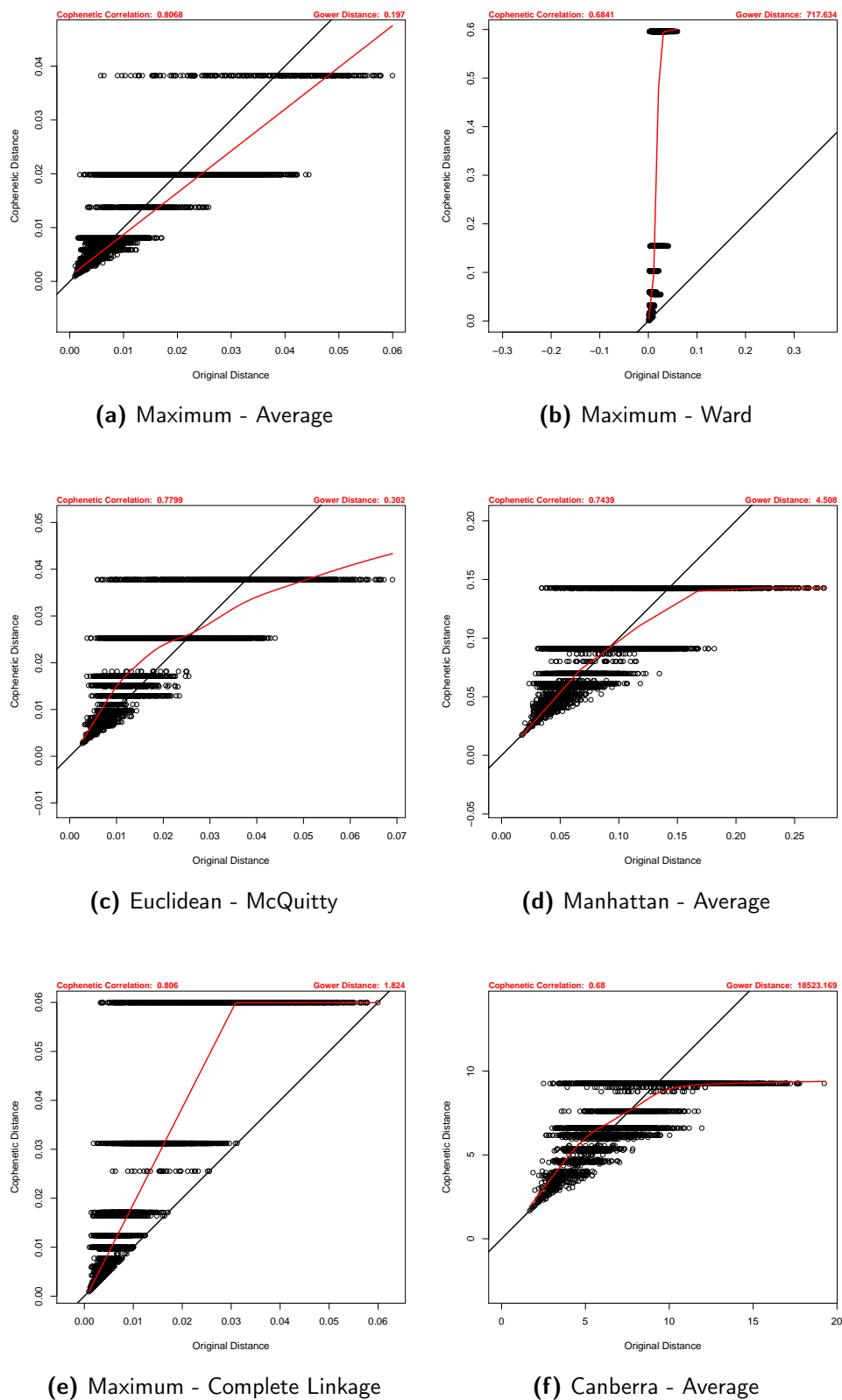
methods, and *Ward's* method gives the best results for all metrics in comparison to the other methods. The second best method, obtained by the *Euclidean* distance metric for the same agglomerative method, has *agglomerative coefficient* 0.991, which is very close to the best method's value. The *banner* plot for the selected partition, *Maximum - Ward*, can be seen in Figure 7.2. The levels of merges have been rescaled to 0-1 to allow for the calculation of the agglomerative coefficient. The labels in the y-axis of the banner are the original identifiers of the 97 patients in the epilepsy data set.

To confirm the method findings, two other statistics will be computed for all 28 methods, i.e. the *Cophenetic correlation* and the *Gower distance* (Borcard et al., 2011). The *Cophenetic correlation* is related to the dendrogram which describes an hierarchical clustering method. More specifically, the *Cophenetic distance* between two items in a dendrogram is defined as the distance at which the two items are joined to the same group. For a pair of items, starting from one of them, climbing up the dendrogram to the first node which leads down to the second item, the level of this node is the *Cophenetic distance* between the two items. Consequently, a *Cophenetic matrix* is a matrix which contains the *Cophenetic distances* between all pairs of items. It is then possible to compute a Pearson's *r* correlation, which is called the *Cophenetic correlation*, between the original dissimilarity matrix of an hierarchical clustering method and the *Cophenetic matrix*. The method with the highest *Cophenetic correlation* can be considered as having the agglomerative method which produced the best clustering method for the distance matrix of the original data. An important aspect of this statistic is that it depends strongly on the clustering method, independently of the data available for analysis. Table 7.4 gives the *Cophenetic correlation* values for all 28 hierarchical clustering methods. Model *Maximum - Average* has the largest *cophenetic correlation* of 0.806. The clustering method *Maximum - Ward* indicated by the *agglomerative coefficient* has *cophenetic correlation* 0.684. Therefore according to this statistic the appropriate method seems to be the former. The relationship between a distance matrix and a *Cophenetic matrix* can be illustrated by means of a *Shepard* like diagram (such as Figure 7.3), plotting the original distances against the *Cophenetic distances* (Legendre and Legendre, 1998). Figure 7.3 illustrates this relationship for six methods for comparison purposes. The *Cophenetic correlation* values of these methods are shown in bold in Table 7.4. More specifically, the methods are the *Maximum - Av-*



**Figure 7.2:** Banner plot for the 2-clustering partition derived by Ward's method using the Maximum distance metric. Height corresponds to the level of merge for a pair of observations, while the labels in the y-axis of the plot are the original identifiers of the patients in the data set.





**Figure 7.3:** Shepard-like diagrams comparing six *Cophenetic distances* to original distances. The Lowess smoothers (red lines) show the trend in each plot. The diagonal (black) lines are visual references.

**Table 7.4:** Pearson's  $r$  *Cophenetic correlation* for the 28 hierarchical clustering methods. In bold are shown the six clustering methods selected using as criteria the *agglomerative coefficient*, *cophenetic correlation* and a combination of these statistics, with their *metric* and *agglomerative method* (see the accompanying text for the reasoning).

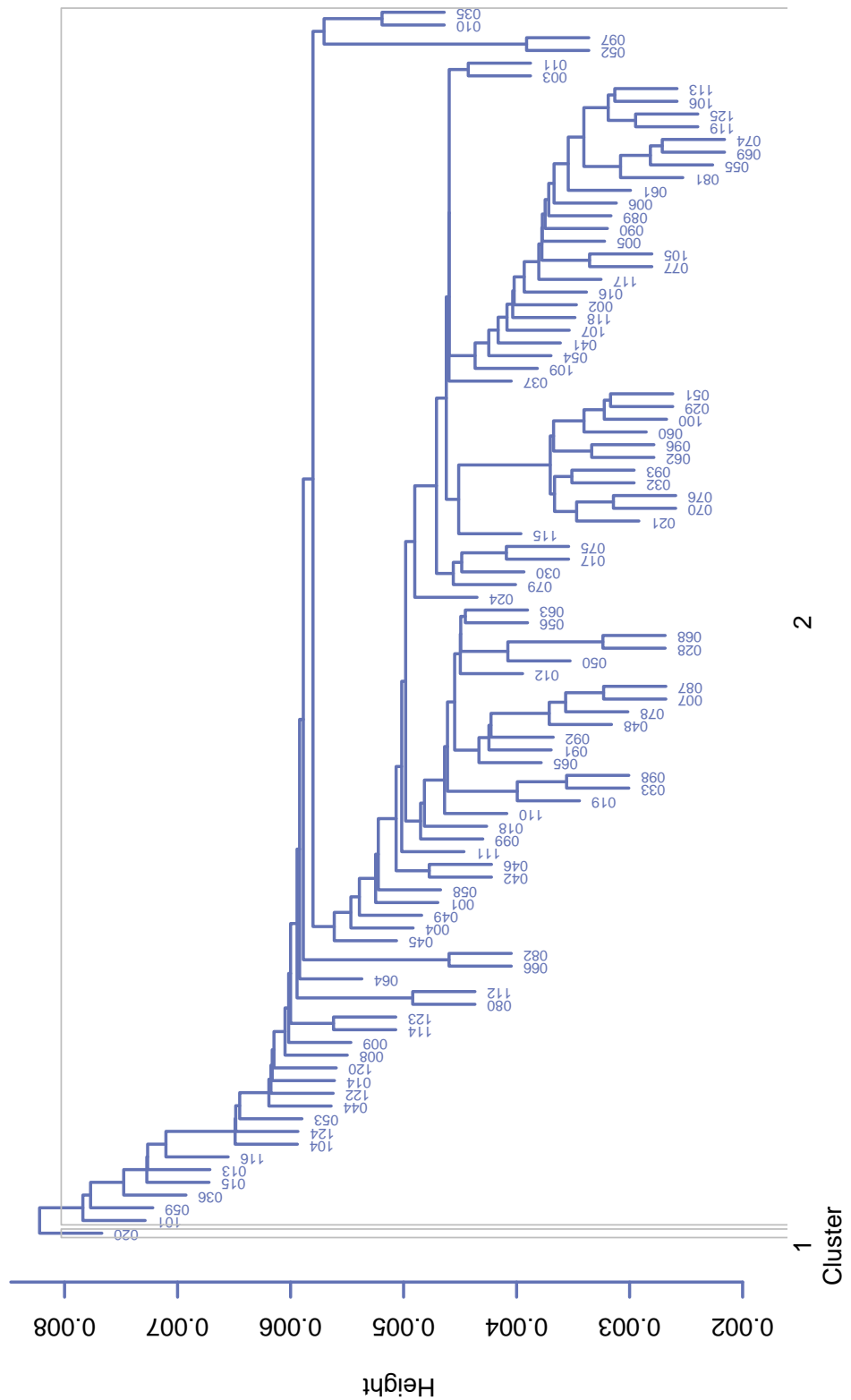
Metric	Agglomerative Method						
	Single	Complete	Average	Ward	McQuitty	Median	Centroid
<b>Euclidean</b>	0.609	0.669	0.755	0.738	<b>0.779</b>	0.741	0.723
<b>Manhattan</b>	0.504	0.708	<b>0.743</b>	0.582	0.703	0.676	0.711
<b>Maximum</b>	0.678	<b>0.805</b>	<b>0.806</b>	<b>0.684</b>	0.716	0.710	0.798
<b>Canberra</b>	0.526	0.631	<b>0.680</b>	0.622	0.657	0.510	0.560

erage having the largest *cophenetic correlation*, the *Maximum - Ward* with the largest AC value, the *Euclidean - McQuitty* with the largest *Cophenetic correlation* among all seven *Euclidean* methods, the *Manhattan - Average* having the largest *Cophenetic correlation* among all seven *Manhattan* methods, the *Maximum - Complete linkage* with the largest *Cophenetic correlation* among all four *Complete linkage* methods, and the *Canberra - Average* method having the largest *Cophenetic correlation* among all seven *Canberra* methods. No *Single linkage* or *Median* and *Centroid* methods were chosen. The former showed extensive chaining in the dendrograms of all such methods, making it difficult to interpret the solution independently of the low values of both the AC and *Cophenetic correlation* statistics e.g. Figure 7.4 for the dendrogram of the *Euclidean - Single linkage* method. The latter two types of methods introduced a lot of crossovers of branches in the corresponding dendrograms, making it very difficult to identify any patterns in their clustering solutions to the problem e.g. Figure 7.5 for the dendrogram of the *Maximum - Centroid* method.

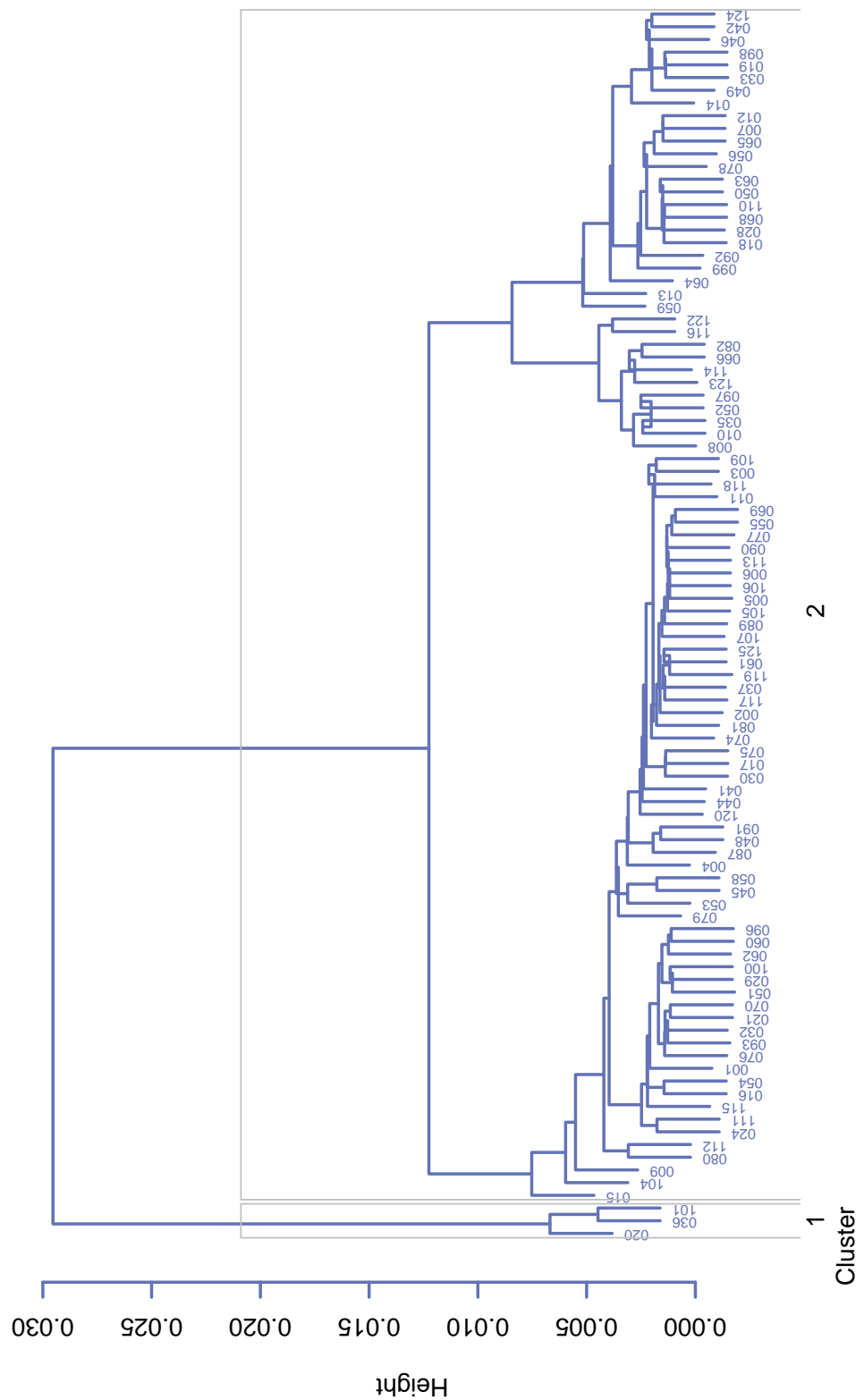
In the *Shepard*-like diagrams of Figure 7.3 there is another measure of the goodness-of-fit between the matrices, the *Gower distance*. This statistic is defined as the sum of squared differences between the values in the two matrices (Legendre and Legendre, 1998). That is,

$$D_{Gower} = \sum_{i,j} (\text{original } d_{ij} - \text{cophenetic } d_{ij})^2.$$

The smaller the value of this statistic, the better the fit of the method to the original data. Similarly to the *Cophenetic correlation*, the *Gower distance* requires the results for comparison to be from the same original distance matrix. In addition, it is not necessary that both statistics indicate the same clustering method as the best. The lowess function that was used to draw the smoothers (red lines) in the diagrams of Figure 7.3 required extensive experimentation with two of its arguments (the *smoother span*,  $f$ , corresponds to the proportion of points in the plot that influence the smoothness at each value, and *delta*, determines the range of points around the last computed point for which the local polynomial fit will not be computed). Each smoother line in the six diagrams was drawn with different values of these two arguments, to allow the line to



**Figure 7.4:** Dendrogram for the 2-cluster partition derived by the *Euclidean - Single linkage* clustering method. The labels at the end-leaves of the tree are the original identifiers of the patients in the data set.



**Figure 7.5:** Dendrogram for the 2-cluster partition derived by the *Maximum - Centroid* clustering method. The labels at the end-leaves of the tree are the original identifiers of the patients in the data set.

approximate as accurately as possible the trend of the points in the plots.

The *Gower distance* values for all available clustering methods can be seen in Table 7.5. The *Average* method is clearly the best among all agglomerative methods in all

**Table 7.5:** *Gower distance* for the 28 hierarchical clustering methods. In bold is shown the clustering method with the smallest *Gower distance* value.

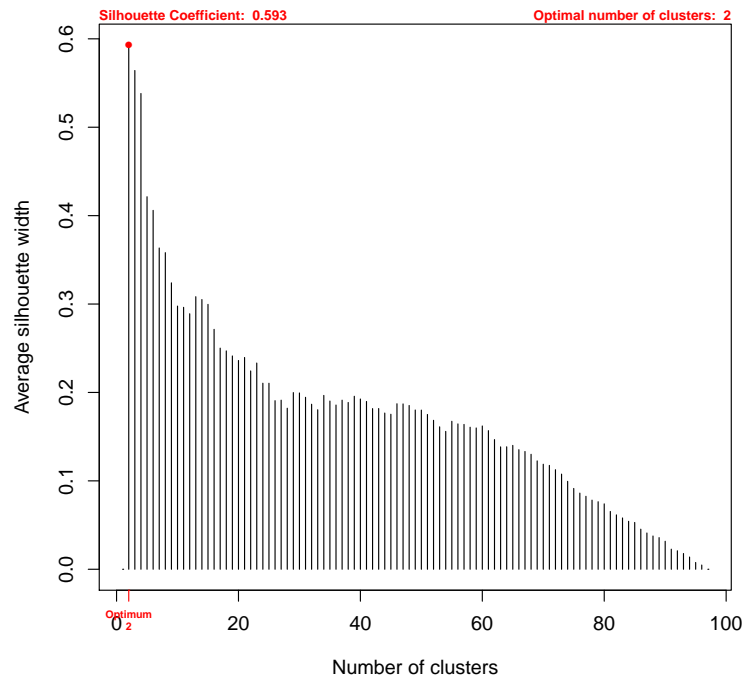
Metric	Agglomerative Method						
	Single	Complete	Average	Ward	McQuitty	Median	Centroid
<b>Euclidean</b>	1.467	4.697	<b>0.294</b>	642.73	0.301	0.530	0.695
<b>Manhattan</b>	23.475	40.119	<b>4.508</b>	12348.73	10.277	11.451	14.761
<b>Maximum</b>	1.120	1.824	<b>0.197</b>	717.634	0.720	0.518	0.360
<b>Canberra</b>	97245.42	228031.3	<b>18523.17</b>	24316750	46636.64	85383.48	93594.2

metrics, as the *Gower distance* value for this method is the smallest in comparison to all other methods and in each and every metric. Therefore, the *Average* method seems to be the most appropriate, with respect to the *Gower distance* results. As the *Maximum - Average* method had the highest *Cophenetic correlation* (as seen on Table 7.4), it seems so far that the best fit to the original data is given by the *Maximum - Average* method. The *Shepard*-like diagram for method *Maximum - Ward* clearly indicates that, despite having the highest *agglomerative coefficient*, it does not fit the original data well. However, further investigation is needed to confirm which of the two *Maximum* methods, *Average* and *Ward*, gives the best-fit of the data, with respect to the *response to AEDs* information.

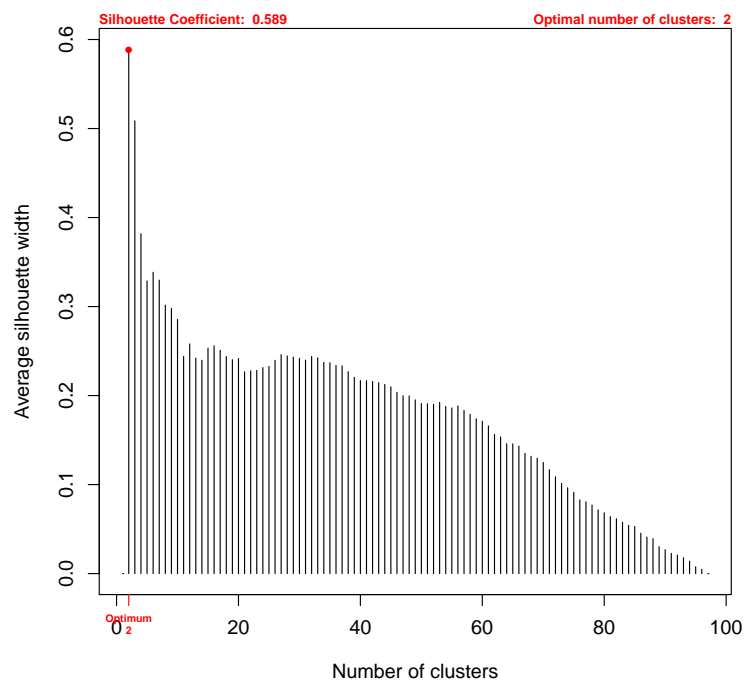
### 7.5.4.3 Identification of the Optimal Number of Clusters

An important part of the clustering procedure is to decide at what level to cut the dendrogram of a clustering solution. This decision can be taken either subjectively by choosing the number of clusters from visual inspection of the dendrogram, or such chosen that it satisfies some criteria. *Silhouette widths* and plots of the *fusion level* values are two methods which can be used to define criteria for the appropriate number of clusters.

As has already been described in detail in Section 7.4, the *silhouette width* is a measure of the degree of membership of an item to its cluster. This measure can be computed and the obtained values drawn in a bar plot for all possible numbers of clusters in a clustering solution. The R function `silhouette()` of package `cluster` will be used to obtain such a plot for the clustering solutions of the epilepsy data. This plot will be drawn for the two clustering methods being discussed, namely the *Maximum - Average* and the *Maximum - Ward*, which have been identified as those methods with most potential as the best-fit method of the original distance matrix. Figure 7.6 illustrates the *average silhouette widths* for all partitions, from 2-96 clusters, for the two clustering methods mentioned previously. It is clear that in both methods the optimal number of



(a) Maximum - Average



(b) Maximum - Ward

**Figure 7.6:** Average silhouette widths for partitions of 2-96 clusters for the two selected clustering methods. The optimal number of clusters is indicated in red.

clusters is 2. In fact, this is true for all 28 clustering methods.

To confirm the findings from *silhouette widths* another type of graph can be used. The *fusion level* values of a dendrogram can be plotted, and from this plot the optimal number of clusters can be identified. A *fusion level* value is the distance at which a fusion between two branches of a dendrogram occurs. Figure 7.7 shows the fusion level values corresponding to the dendrograms of the two clustering methods. Reading the graphs from right to left (2 clusters to 97 clusters), it can be seen that in both methods there is a large jump after the two-clusters fusion. Especially in the case of the *Maximum - Ward* graph, this is even more pronounced. Therefore, both plotting tools indicate that the optimal number of clusters is 2 for both clustering methods.

As the optimal number of clusters is now known, it is possible and useful to compare the cluster contents among the dendrograms by means of contingency tables. Table 7.6 shows the comparison of the classifications obtained by the two clustering methods. The

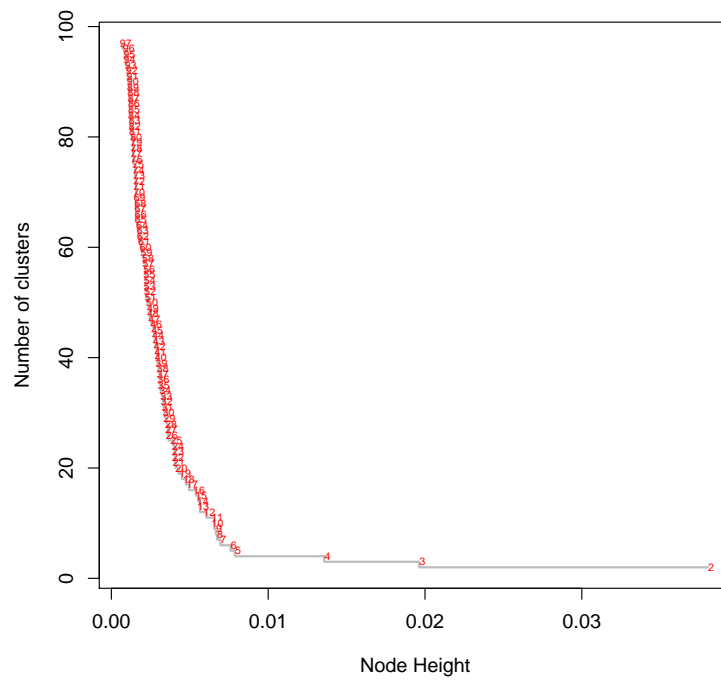
**Table 7.6:** Cross-tabulation of the 2-cluster partitions for *Response to AEDs*.

		Maximum - Average	
		1	2
Maximum - Ward	1	63	0
	2	31	3

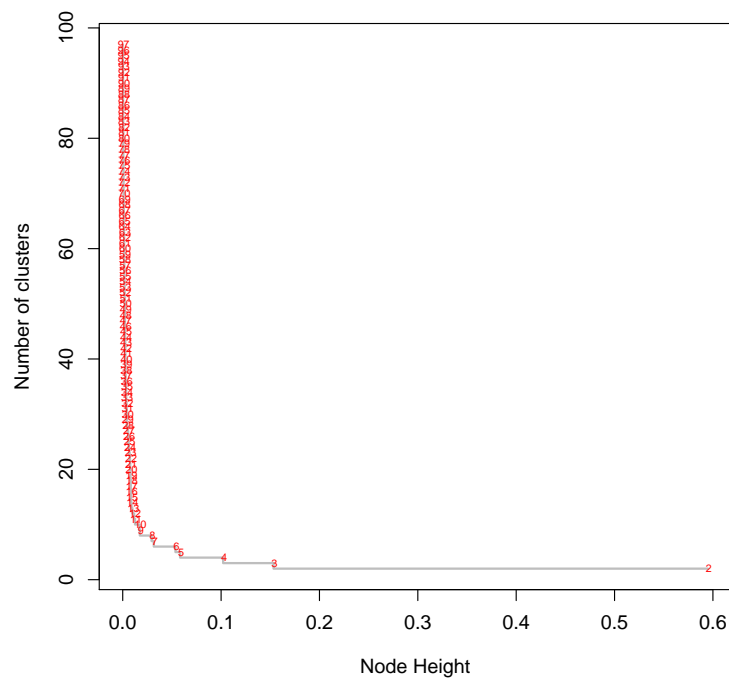
contingency tables show that *Maximum - Average* has classified all but three patients to the first cluster, whereas *Maximum - Ward* has classified 63 patients to cluster 1 and 31 to cluster 2. This could be important with respect to the ability of these clustering methods to discriminate between responders and non-responders to AEDs.

#### 7.5.4.4 Identification of the Best Method for the *Response to AEDs* Information

Contingency tables can also be used to compare the cluster contents with respect to a clinical characteristic of the patients in the data, such as the *Response to AEDs*. Table 7.7 contains the clustering information for the two clustering methods with respect to *Response to AEDs*. It is clear that the results for *Maximum - Ward* are far more balanced than those of *Maximum - Average* with respect to the *Response to AEDs*. This is not surprising, as it was already known from Table 7.6 that the latter method classified 94 patients in the first cluster, therefore it could not be possible to separate the patients with regards to their *Response to AEDs* using this method. In addition, Table 7.7 indicates that the method *Maximum - Ward* has misclassified  $17 + 28 = 45$  patients with respect to *Response*, while the *Maximum - Average* method misclassified



(a) Maximum - Average



(b) Maximum - Ward

**Figure 7.7:** Graphs of the *fusion level* values of the corresponding dendrograms to the two clustering methods. The numbers in red are the number of clusters obtained at specific node heights.



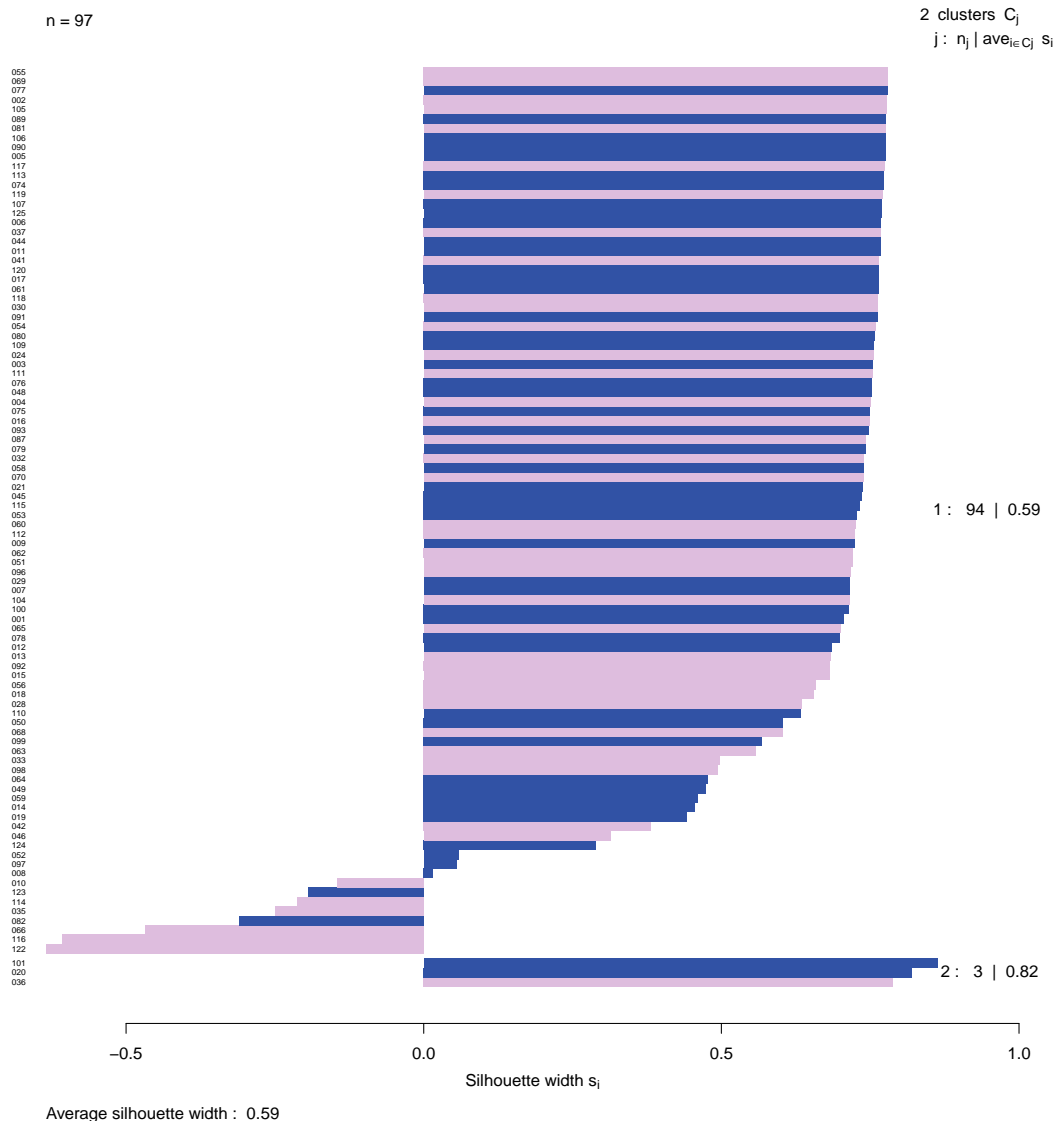
**Table 7.7:** Cross-tabulation of 2-cluster partition for the two methods to the *Response to AEDs* clinical characteristic.

	Response to AEDs		
		Responder	Non-responder
Maximum - Average	1	50	44
	2	2	1
Maximum - Ward	1	35	28
	2	17	17

$2 + 44 = 46$  patients<sup>1</sup>. Hence, the former method seems to be very slightly more accurate with respect to the *Response* information. However, both methods clearly are not capable of discriminating, with a small misclassification error the patients with regards to their *Response* information. This will become more obvious with the aid of a number of plotting tools, to illustrate these findings graphically.

Although dendrograms (e.g. Figure 7.11) and heat maps (e.g. Figure 7.12) illustrate the clustering result achieved by the application of a clustering method to the data, another type of graphical tool, the *silhouette plot* (based on the *silhouette widths*) can show how well each and every patient has been assigned to its respective cluster after the classification process, i.e. That is, to what degree a patient is a member of its cluster. The silhouette plots for the two clustering methods can be seen in Figures 7.8 and 7.9 for the *Maximum - Average* and *Maximum - Ward* methods respectively. All patients' *silhouette width* values can be seen in the *silhouette plot* as bars, ranked in decreasing order, with the colours of these bars corresponding to the *Response to AEDs* information, e.g. responders to AEDs are depicted with blue bars and non-responders with pink bars. It is therefore, clear, which patients lie well within their cluster. The wider the *silhouette* bar for a patient is, the larger the *silhouette* value for this patient and the better the patient lies in the cluster. That is, the within cluster dissimilarity of the patient is much smaller than the smallest dissimilarity of the patient to other clusters. Both clustering methods have *Average Silhouette width* for the entire data of 0.59 (which is also the *Silhouette coefficient* for both methods), therefore there is no difference between them with respect to the overall data set. According to Table 7.1, a reasonable amount of clustering structure has been discovered by the classification algorithm that was used. On the other hand, the *Average Silhouette widths* for the clusters differ considerably, as the number of patients in the two clusters are not the same as has already been seen from the contingency tables. In addition, the *silhouette plot* for method *Maximum - Average* shows that there are 8 patients clearly misclassified (highly negative *Silhouette widths*) as members of cluster 1, while according to the *silhouette*

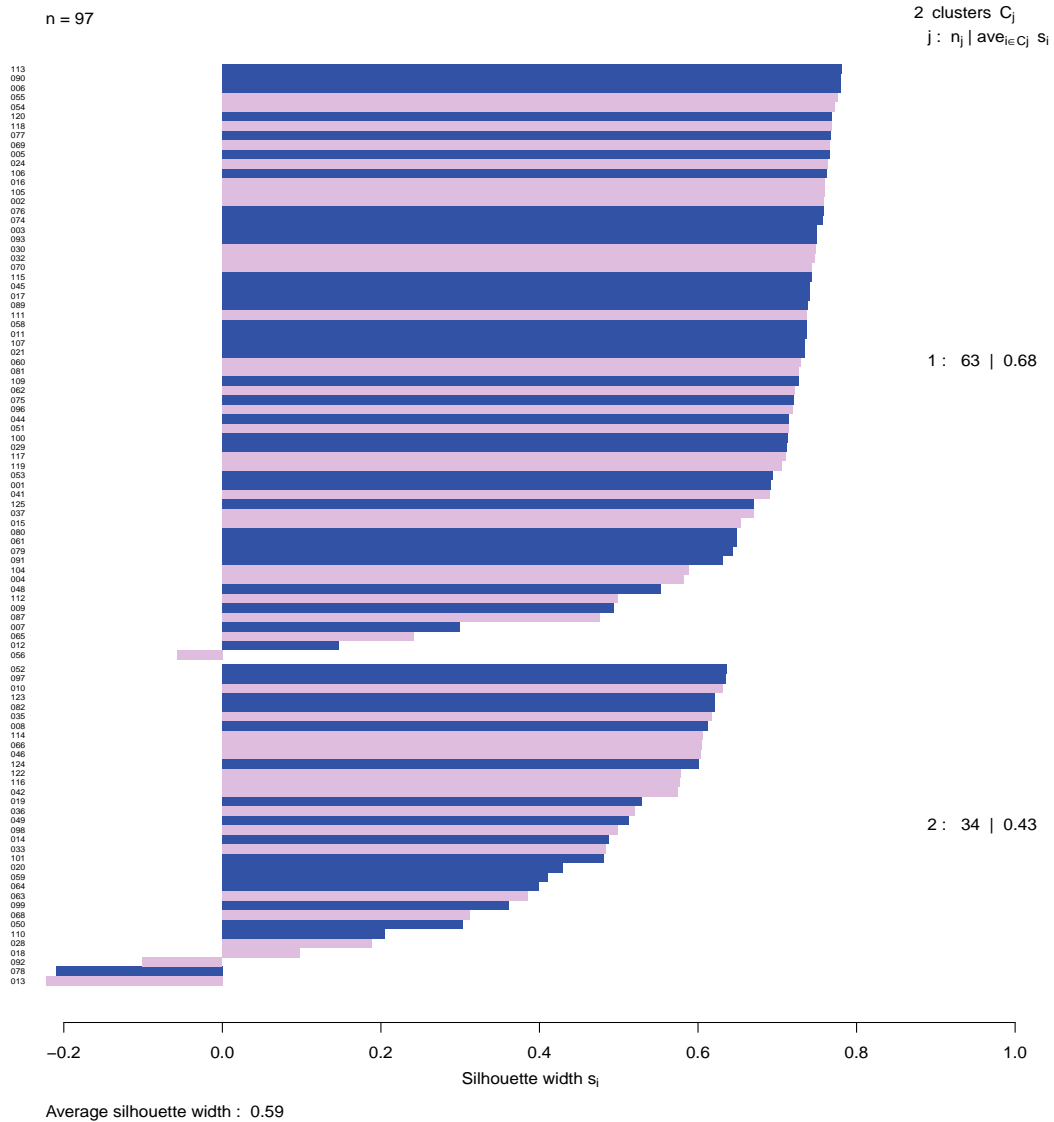
<sup>1</sup>It should be noted that the results of the cross-tabulation of the 2-cluster partitions to *Response to AEDs*, were not improved, when computed by all other 26 clustering methods, hence these were the best results among the available methods.



**Figure 7.8:** *Silhouette plot* for the 2-cluster partition derived by the *Maximum - Average* clustering method. The blue and the pink bars correspond to responders and non-responders respectively. The *average silhouette width* for clusters 1 and 2 is 0.59 and 0.82 respectively, and the *average silhouette width* for the entire data set is 0.59.

*plot* these should have been members of cluster 2. Model *Maximum - Ward* is definitely more balanced, with only 1 patient misclassified in cluster 1 (actually belonging to cluster 2) and 3 patients in cluster 2 (belonging to cluster 1) having in general far smaller negative *silhouette widths* than the misclassified patients in method *Maximum - Average*. The findings and the information obtained by the *silhouette plots* practically mean that only the *Maximum - Ward* method should be retained for further analyses, as it is deemed to be the best with regards to the *Response to AEDs* information.

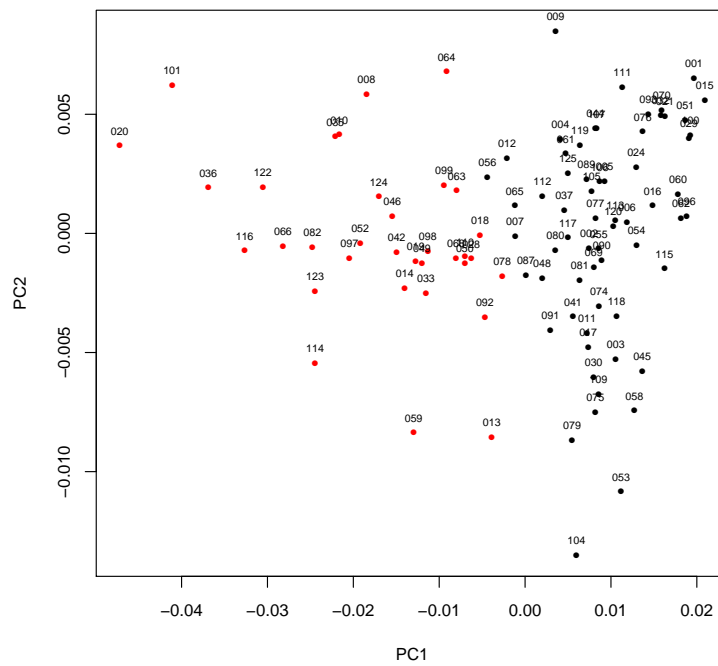
To illustrate the clustering solution derived by the *Maximum - Ward* method, a number of graphical tools will be used. A two-dimensional projection of the clustering



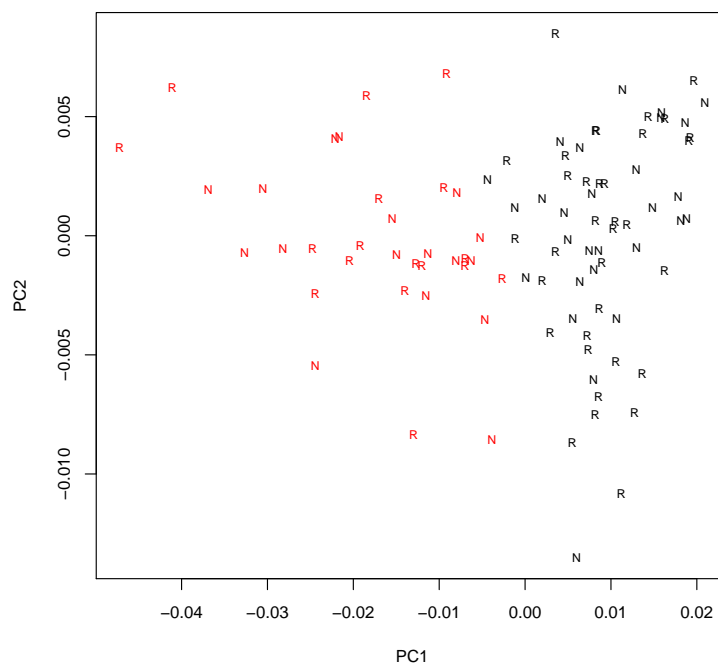
**Figure 7.9:** *Silhouette plot* for the 2-cluster partition derived by the *Maximum - Ward* clustering method. The blue and the pink bars correspond to responders and non-responders respectively. The *average silhouette width* for clusters 1 and 2 is 0.68 and 0.43 respectively, and the *average silhouette width* for the entire data set is 0.59.

solution can be seen in Figure 7.10. The first two principal component scores (according to the results from Chapter 5) can be seen superimposed with the partition derived by Ward's method. In both scores plots, black and red represent the patients clustered to the first and second cluster respectively. The bottom scores plot illustrates the *Response to AEDs* information, such that points labelled as "R" and "N" correspond to responders or non-responders to AEDs respectively. The scores plots show clearly that there is no discrimination between responders and non-responders to AEDs.

A dendrogram for the 2-clustering partition derived by the *Maximum - Ward* clustering method can be seen in Figure 7.11. The labels at the end-leaves of the tree correspond



(a) 2-cluster partition



(b) Response to AEDs

**Figure 7.10:** Scores plots of the first two PCs, superimposed with the 2-cluster partition derived by the *Maximum - Ward* clustering method. Black and red points represent the patients in the first and second cluster respectively. The labels of the points in the bottom plot correspond to the responders (R) and non-responders (N) to AEDs.

to the responders (R) and non-responders (N) to AEDs. As expected from the results obtained so far, there are indeed two main clusters fused at a very high level of height of 0.595, compared to all other fusions (with the next highest merge of patients or clusters occurring at height 0.154). The dendrogram is also quite balanced.

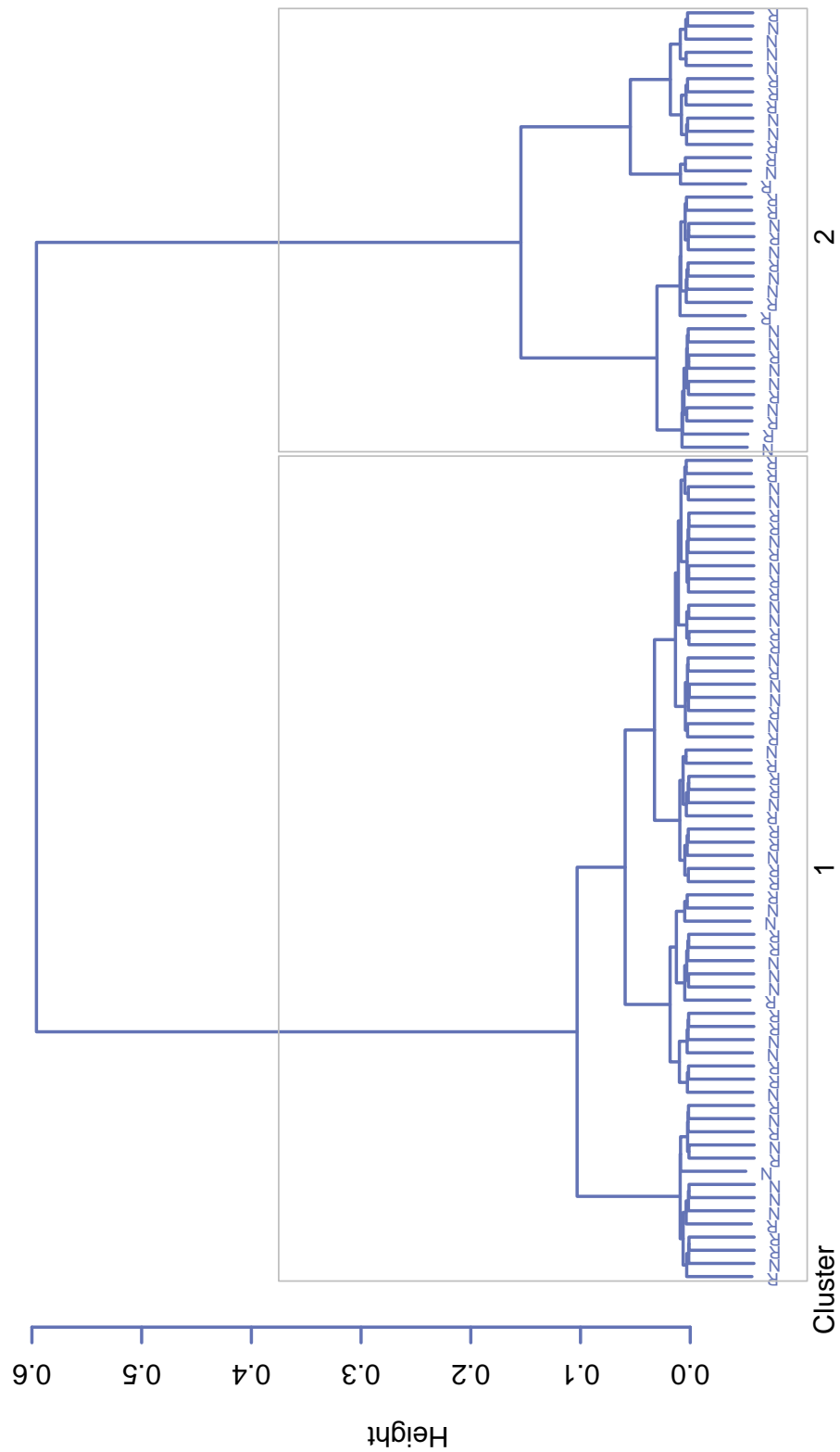
A dendrogram can also be represented, perhaps more accurately, by a heat map, a square matrix of coloured pixels such that the colour intensity represents the similarity among the patients. The heat map of the distance matrix reordered according to the dendrogram of Figure 7.11 can be seen in Figure 7.12. The reordering of the *heat map* sorts the matrix such that most of the darker (or red) values representing high similarities are located closer to the main diagonal. The heat map shows the two large dark red square areas at the top-left and bottom-right of the matrix, corresponding to the two clusters identified from previous analyses.

Despite the optimal number of clusters being 2, it might be useful to examine whether partitions of larger number of clusters than 2 can provide an insight to the discrimination of the *Response* information. Table 7.8 shows the results of cross-tabulating the *Response* information with the partitions of 3-6 clusters, obtained from clustering method *Maximum - Ward*. As can be seen, no cluster in all four partitions in Table

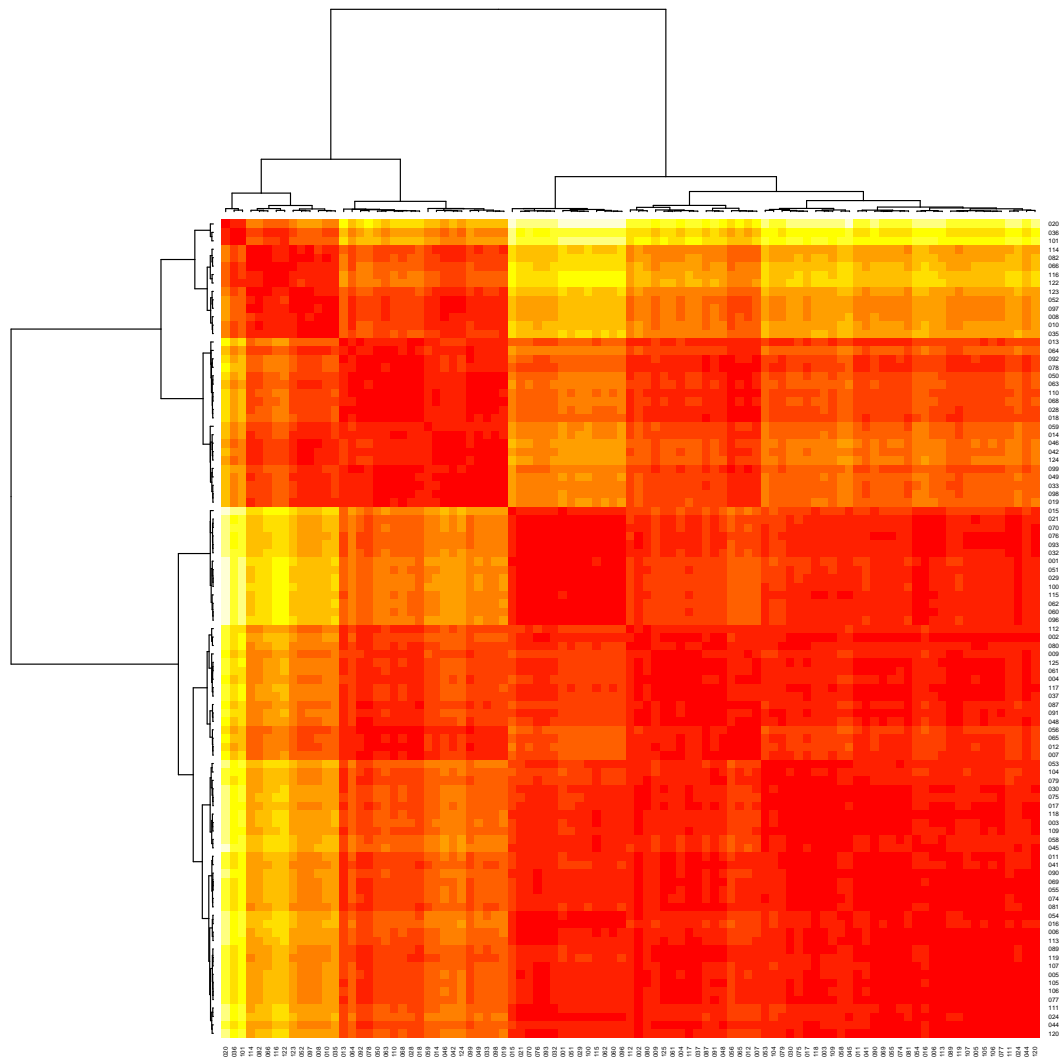
**Table 7.8:** Cross-tabulation of 3-6 cluster partitions for the *Maximum - Ward* method with the *Response to AEDs* clinical characteristic. In bold are depicted the clusters that are not affected by the introduction of new clusters in the 3-cluster partition.

Clusters	Response to AEDs		
	Responder	Non-responder	
3	<b>1</b>	35	28
	<b>2</b>	<b>7</b>	<b>7</b>
	<b>3</b>	<b>10</b>	<b>10</b>
4	1	7	7
	2	28	21
	<b>3</b>	<b>7</b>	<b>7</b>
	<b>4</b>	<b>10</b>	<b>10</b>
5	1	7	7
	2	8	8
	<b>3</b>	20	13
	<b>4</b>	<b>7</b>	<b>7</b>
	<b>5</b>	<b>10</b>	<b>10</b>
6	1	7	7
	2	8	8
	<b>3</b>	20	13
	4	5	6
	<b>5</b>	<b>10</b>	<b>10</b>
	<b>6</b>	2	1

7.8 contains only one of the two categories of response to AEDs. In addition, it seems that the last two clusters in the 3-cluster partition (of size 14 and 20 respectively) are



**Figure 7.11:** *Dendrogram* for the 2-cluster partition derived by the *Maximum - Ward* clustering method. The labels at the end-leaves of the tree correspond to the responders (R) and non-responders (N) to AEDs.



**Figure 7.12:** Heat map of the distance matrix of the *Maximum - Ward* clustering method according to the dendrogram of Figure 7.11. The colour intensity represents the similarity among the patients, such that the darker the colour the closer the similarity.

not separated when 1 or 2 more clusters are introduced, and the cluster of size 20 also remains unchanged even in the 6-cluster partition. Most of the clusters in all partitions are balanced with regards to *Response to AEDs*, having the same number of responders and non-responders. In the 6-cluster partition, cluster 6 contains just one more responder than non-responder, whereas cluster 4 is the only cluster among all clusters in all partitions which contains more non-responders than responders. Only the largest cluster shows any real difference between the two responses, with the responders dominating the cluster, with sizes 35 and 28 in 3-clusters, 28 and 21 in 4-clusters, 20 and 13 in 5 and 6-clusters, for responders and non-responders respectively.

To investigate the homogeneity of the four partitions in Table 7.8 with respect to the observations in each cell for the two categories of *Response to AEDs*, the  $\chi^2$  test will be used. The  $p$ -values for the five (including the 2-cluster partition) cases can be seen

in Table 7.9. All  $p$ -values are much larger than the significance level of 0.05, therefore

**Table 7.9:**  $\chi^2$  test for homogeneity of the clusters with respect to the proportion of observations in each of the categories which the *Response to AEDs* is divided into. The  $p$ -value for the 6-cluster partition has been computed with Fisher's test, as there are expected frequencies of value  $< 5$  in at least one of the cells of the respective contingency table.

Clusters	$P$ -value
2	0.7565
3	0.8719
4	0.9195
5	0.9121
6	0.9168

there is not enough evidence to reject the null hypotheses of the tests that the clusters in all partitions are not homogeneous with respect to the *Response to AEDs*. That is, the proportions of observations in each of the categories of *Response to AEDs* are not different in the five derived partitions (2-6 clusters).

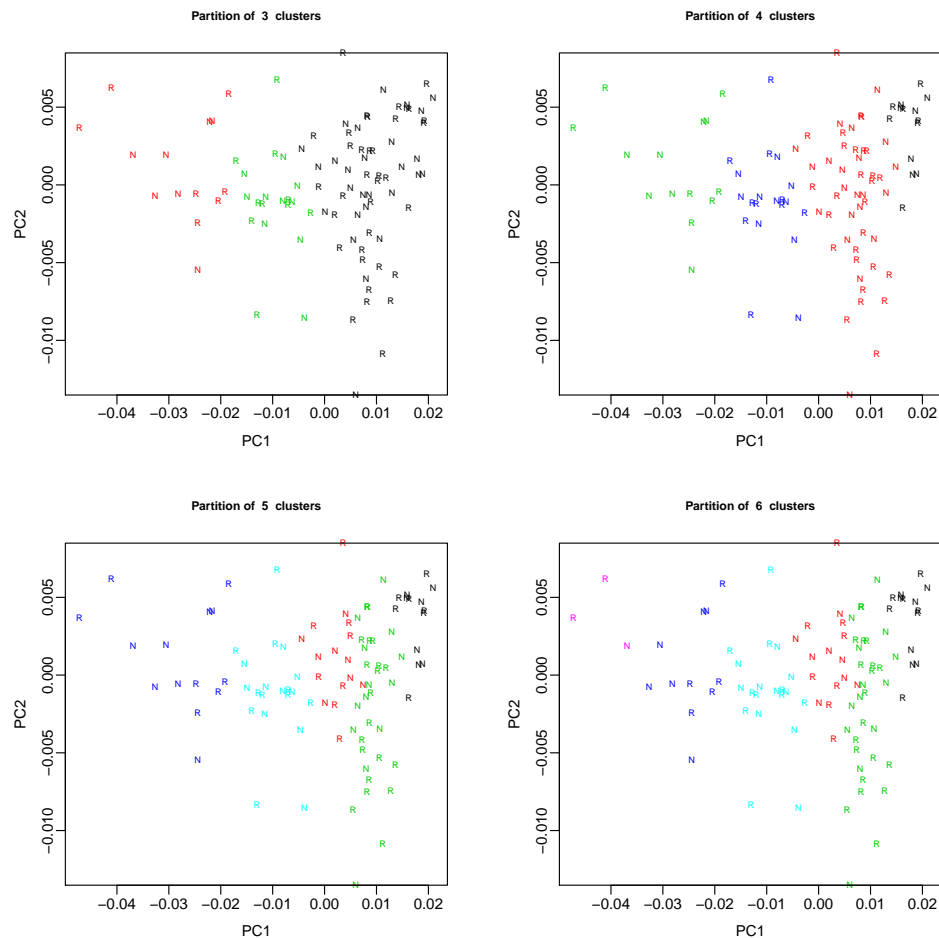
Figure 7.13 illustrates the results derived from the contingency tables (Table 7.8) for the *Response*. The colours of the points in the scores plots correspond to the clusters in each partition. In the 3-cluster partition, the left-most cluster (red points) is the cluster of size 14, which remains unchanged up to and including the 5-clusters partition, and the middle cluster (green points) is the cluster of size 20, which remains unchanged in all four partitions. The right-most cluster (black points) is the largest cluster in all partitions, which keeps being broken into smaller partitions until the 6-cluster partition, where the left-most partition is broken for the first time. Figure 7.13 shows clearly that there is no discrimination between responders and non-responders in any of the four partitions.

As a conclusion to these analyses, it can be said that the hierarchical clustering methods have not been efficient in classifying the patients according to their *Response to AEDs*. In general, it can be seen from the scores plots that the clusters are more elongated than compact and the distance between them is not great, in most of the cases.

#### 7.5.4.5 Investigation on Other Clinical Characteristics

*Response to AEDs* is not the only clinical characteristic of the patients in the epilepsy data. It might be useful to assess whether the hierarchical clustering method *Maximum - Ward* discussed in the previous sections can provide a better insight to the clusters of the derived partitions than for the *Response to AEDs*. Four main characteristics will be used in the analyses, namely *Gender*, *Age*, *BMI* and *Seizure Type*. The *Age* and *BMI* categories are those defined in Chapter 2. Table 7.10 contains the contingency tables between the 2-6 cluster partitions and the clinical characteristics *Gender* and *Age*.





**Figure 7.13:** Scores plots of the first two PCs, superimposed with the cluster partitions for 3-6 clusters, derived by the *Maximum - Ward* clustering method. The labels of the points in the plots correspond to the responders (R) and non-responders (N) to AEDs.

The cross-tabulation of *Gender* shows that, as expected, due to their large number in the data set, males dominate most of the clusters in all partitions, with practically the only exception being cluster 3 in the last two partitions in the table. The third cluster in the 3-cluster partition (indicated in bold in the table), of size 20 (13 males and 7 females) is retained as it is, in the following three partitions, albeit in the last two as cluster 5. Contrary to *Response to AEDs*, in this case there is one cluster that contains only males, the sixth cluster in the last partition, and a few other clusters with very high numbers of males compared to females, e.g. cluster 1 in the 3-5 cluster partitions contains 13 males and 1 female.

To investigate the homogeneity of the five partitions in Table 7.8 with respect to the observations in each cell for the two categories of *Gender*, the  $\chi^2$  test will be used. The  $p$ -values for the five cases can be seen in Table 7.11. The  $p$ -values for the first three partitions are larger than the significance level of 0.05, therefore there is not enough evidence to reject the null hypotheses of the tests that the clusters in these partitions are

**Table 7.10:** Cross-tabulation of 2-6 cluster partitions derived by the *Maximum - Ward* method with the *Gender* and *Age* clinical characteristics. In bold are shown the clusters that are not affected by the introduction of new clusters in the 3 or 4-cluster partition.

Clusters	Gender		Age			
	Male	Female	(16-26]	(26-47]	(47-99]	
2	<b>1</b>	41	22	27	19	17
	<b>2</b>	24	10	4	13	17
3	<b>1</b>	41	22	27	19	17
	<b>2</b>	11	3	0	8	6
	<b>3</b>	<b>13</b>	<b>7</b>	<b>4</b>	<b>5</b>	<b>11</b>
4	<b>1</b>	<b>13</b>	<b>1</b>	<b>8</b>	<b>4</b>	<b>2</b>
	<b>2</b>	28	21	19	15	15
	<b>3</b>	11	3	0	8	6
	<b>4</b>	<b>13</b>	<b>7</b>	<b>4</b>	<b>5</b>	<b>11</b>
5	<b>1</b>	<b>13</b>	<b>1</b>	<b>8</b>	<b>4</b>	<b>2</b>
	<b>2</b>	12	4	5	7	4
	<b>3</b>	16	17	14	8	11
	<b>4</b>	11	3	0	8	6
	<b>5</b>	<b>13</b>	<b>7</b>	<b>4</b>	<b>5</b>	<b>11</b>
6	<b>1</b>	<b>13</b>	<b>1</b>	<b>8</b>	<b>4</b>	<b>2</b>
	<b>2</b>	12	4	5	7	4
	<b>3</b>	16	17	14	8	11
	<b>4</b>	8	3	0	6	5
	<b>5</b>	<b>13</b>	<b>7</b>	<b>4</b>	<b>5</b>	<b>11</b>
	<b>6</b>	3	0	0	2	1

non homogeneous with respect to the *Gender*. That is, the proportions of observations in each of the categories of *Gender* are not different in these 3 partitions. The  $p$ -values for the 5 and 6-cluster partitions are smaller than 0.05 (0.0301 and 0.042 respectively). The clusters in these partitions are not homogeneous with respect to *Gender* as the proportions of observations for the categories of *Gender* are different in at least one of the clusters. From Table 7.10, it can be seen that the third cluster in both the 5 and 6-cluster partitions (containing 16 males and 17 females is the only one with a balanced

**Table 7.11:**  $\chi^2$  test for homogeneity of the clusters with respect to the proportion of observations in each of the categories which *Gender* is divided into. The  $p$ -value for the 6-cluster partition has been computed with *Fisher's* test, as there are expected frequencies of value  $< 5$  in at least one of the cells of the respective contingency table. The statistically significant  $p$ -values at 95% confidence level are shown in bold.

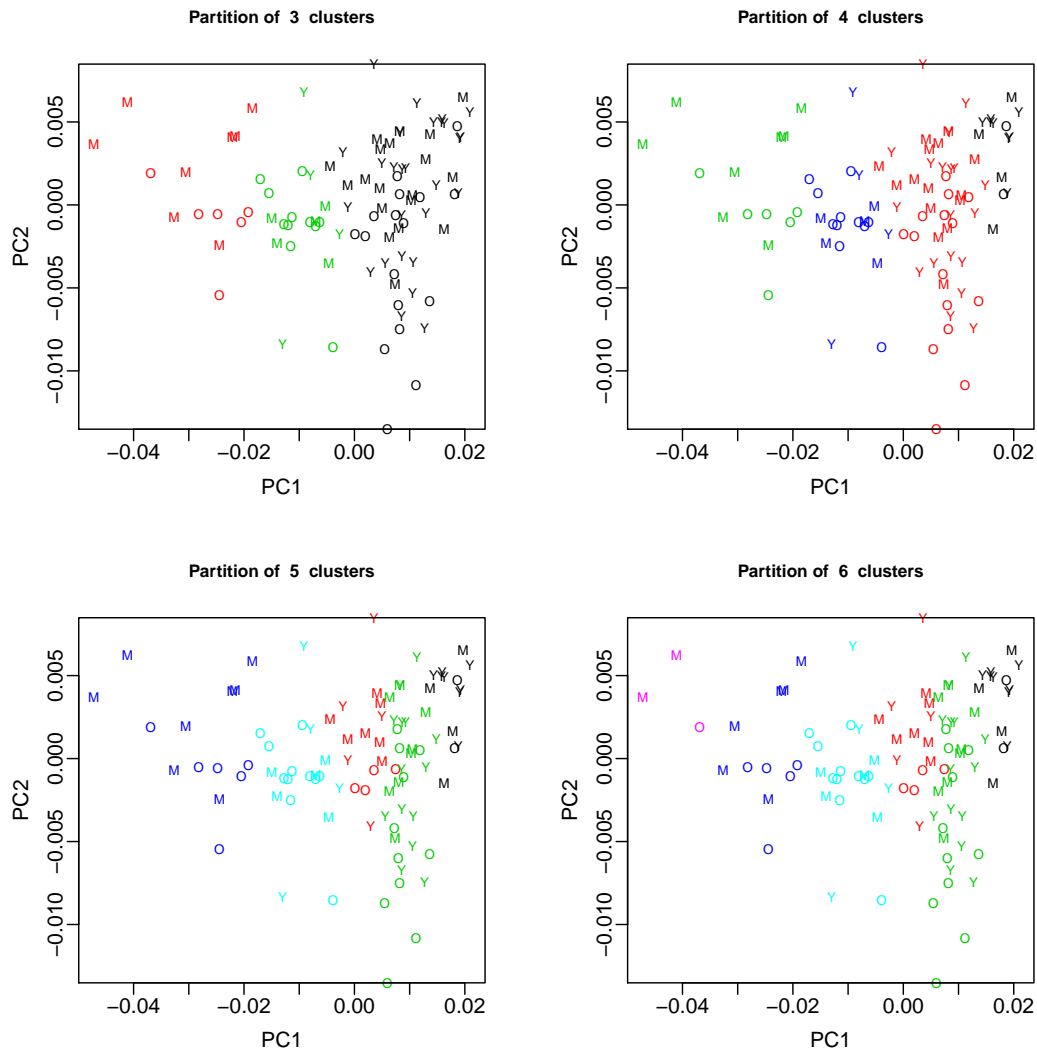
Clusters	$P$ -value
<b>2</b>	0.7457
<b>3</b>	0.6098
<b>4</b>	0.0637
<b>5</b>	<b>0.0301</b>
<b>6</b>	<b>0.0429</b>

number of patients with respect to their gender. This cluster has proportionately more females than we would expect, due to the dominance in number of the males in the data. The rest of the clusters contain more males than females to a ratio of at least 2:1.

Concerning the contingency table for *Age*, cluster 2 in the 3-cluster partition (becoming cluster 3 and 4 in the 4 and 5-cluster partitions respectively), does not contain any patients of age less than or equal to 26, with the patients in this cluster being rather balanced between the other two *Age* intervals. That is, they are not affected by the introduction of new clusters in these three partitions. Two other clusters (containing 4-5-11 and 8-4-2 patients in each *Age* interval respectively), printed in bold in Table 7.10, remain unchanged from the 3 and 4-cluster partitions respectively. The former cluster contains twice as many males as females and their age is above 26 with those of age above 47 dominating this cluster, whereas the latter cluster contains males, the majority of which are of age less than or equal to 26. The cluster with more females than males contains patients of balanced age among the three *Age* levels, with slightly more young and old patients than middle-aged.

Figure 7.14 illustrates the results derived from the contingency tables for *Age*. The colours of the points in the scores plots correspond to the clusters in each partition. In the 3-cluster partition, the left-most cluster (red points) is the cluster of size 14 with only middle-aged or old patients, and remains unchanged in the 3-5 cluster partitions. The middle cluster (green points) is the cluster of size 20, which remains unchanged in all four partitions. This cluster contains mainly old patients in a ratio of approximately 2:1 to the young and middle-aged patients. The right-most cluster (black points) is the largest cluster in all partitions containing patients of all ages (although there are slightly more young patients with a ratio of 4:3), which keeps being divided into smaller partitions until the 6-cluster partition, where the left-most partition is broken for the first time. Things are even clearer in the partitions with larger numbers of clusters. Especially in the 6-cluster partition, cluster 1 is dominated by young patients, whereas in clusters 4 and 6 there are no young patients at all. In addition, clusters 2 and 3 are quite balanced with respect to *Age*. Thus, there is clearly discrimination between the three categories of *Age* in the derived partitions, and the clustering method works in this case, as was also shown by the results of the KW test (7.14).

Table 7.12 contains the contingency tables between the 2-6 cluster partitions and the clinical characteristics *Seizure type* and *BMI*. Concerning *Seizure Type*, all clusters in all partitions are dominated by the LRE patients, as their number in the data set is considerably larger than those patients of IGE type. Similarly to the other previously mentioned clinical characteristics, there are also a couple of clusters here that are very consistent among the various partitions. More specifically, the third cluster in the 3-cluster partition remains unchanged in the rest of the partitions (containing 17 LRE and 3 IGE patients). Also, the first cluster in the 4-cluster partition remains as it is, in



**Figure 7.14:** Scores plots of the first two PCs, superimposed with the cluster partitions for 3-6 clusters, derived by the *Maximum - Ward* clustering method and the *Age* information. The labels of the points in the plots correspond to the young (Y), middle-aged (M) and old (O) patients.

the 5- and 6-cluster partitions (containing 9 LRE and 5 IGE patients).

To investigate the homogeneity of the five partitions in Table 7.12 with respect to the observations in each cell for the two categories of *Seizure Type*, the  $\chi^2$  test will be used. The  $p$ -values for the five cases can be seen in Table 7.13. All  $p$ -values are larger than the significance level of 0.05 therefore there is not enough evidence to reject the null hypotheses of the tests that the clusters in all partitions are not homogeneous with respect to *Seizure type*. That is, the proportions of observations in each of the categories of *Seizure Type* are not different in the 6 derived partitions.

Similarly to the findings for *Seizure Type*, the results for *BMI* show that the same clusters are consistent among the various partitions (containing 4-4-6-6 and 9-4-1-0

**Table 7.12:** Cross-tabulation of 2-6 clusters partitions derived by the *Maximum - Ward* method with the *Seizure Type* and *BMI* clinical characteristics. In bold are shown the clusters that are not affected by the introduction of new clusters in the 3 or 4-cluster partition.

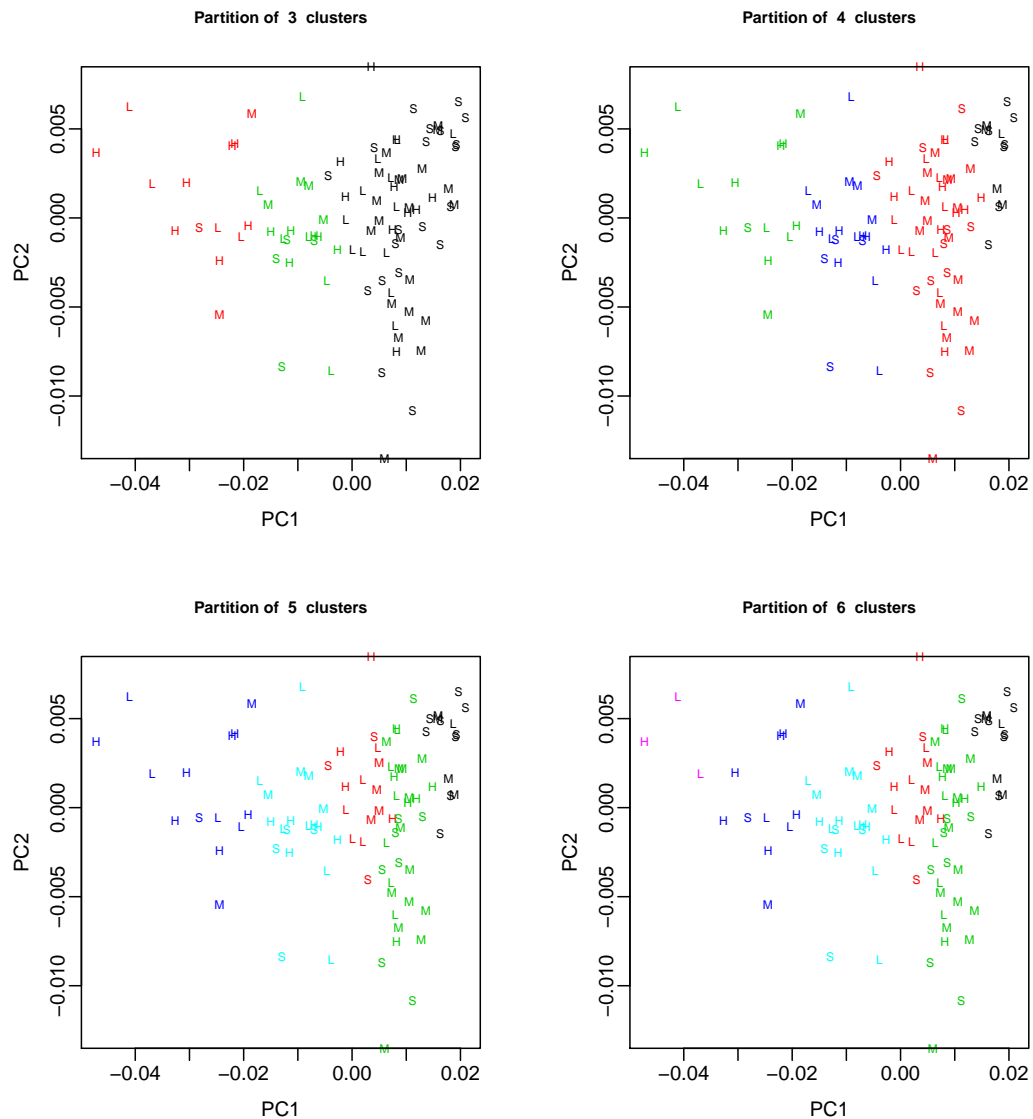
Clusters	Seizure Type			BMI			
	LRE	IGE		(16-22]	(22-25]	(25-28]	(28-45.1]
2	<b>1</b>	45	18	20	21	12	10
	<b>2</b>	30	4	5	6	10	13
3	<b>1</b>	45	18	20	21	12	10
	<b>2</b>	13	1	1	2	4	7
	<b>3</b>	<b>17</b>	<b>3</b>	<b>4</b>	<b>4</b>	<b>6</b>	<b>6</b>
4	<b>1</b>	<b>9</b>	<b>5</b>	<b>9</b>	<b>4</b>	<b>1</b>	<b>0</b>
	<b>2</b>	36	13	11	17	11	10
	<b>3</b>	13	1	1	2	4	7
	<b>4</b>	<b>17</b>	<b>3</b>	<b>4</b>	<b>4</b>	<b>6</b>	<b>6</b>
5	<b>1</b>	<b>9</b>	<b>5</b>	<b>9</b>	<b>4</b>	<b>1</b>	<b>0</b>
	<b>2</b>	12	4	3	4	5	4
	<b>3</b>	24	9	8	13	6	6
	<b>4</b>	13	1	1	2	4	7
	<b>5</b>	<b>17</b>	<b>3</b>	<b>4</b>	<b>4</b>	<b>6</b>	<b>6</b>
6	<b>1</b>	<b>9</b>	<b>5</b>	<b>9</b>	<b>4</b>	<b>1</b>	<b>0</b>
	<b>2</b>	12	4	3	4	5	4
	<b>3</b>	24	9	8	13	6	6
	<b>4</b>	10	1	1	2	2	6
	<b>5</b>	<b>17</b>	<b>3</b>	<b>4</b>	<b>4</b>	<b>6</b>	<b>6</b>
	<b>6</b>	3	0	0	0	2	1

patients in the four *BMI* levels for the same clusters and partitions as for the *Seizure Type*.

Figure 7.15 illustrates the results derived from the contingency tables for *BMI*. The colours of the points in the scores plots correspond to the clusters in each partition. In the 3-cluster partition, the left-most cluster (red points) is the cluster of size 14 which is dominated by patients with *BMI* values greater than 28, and remains unchanged in the 3-5 cluster partitions. The middle cluster (green points) is the cluster of size 20, with balanced *BMI* values, and remains unchanged in all four partitions. The right-

**Table 7.13:**  $\chi^2$  test for homogeneity of the clusters with respect to the proportion of observations in each of the categories the *Seizure Type* is divided into. The *p*-value for the 3-6 cluster partitions have been computed with *Fisher's* test, as there are expected frequencies of value  $< 5$ , in at least one of the cells of the respective contingency table.

Clusters	P-value
<b>2</b>	0.1027
<b>3</b>	0.1757
<b>4</b>	0.2152
<b>5</b>	0.3654
<b>6</b>	0.5673



**Figure 7.15:** Scores plots of the first two PCs, superimposed with the cluster partitions for 3-6 clusters, derived by the *Maximum - Ward* clustering method. The labels of the points in the plots correspond to patients with small (S), medium (M), large (L) and huge (H) *BMI* values.

most cluster (black points) is the largest cluster in all partitions, containing patients of mainly small and medium *BMI* values (in a ratio of approximately 2:1), which keeps being divided into smaller partitions until the 6-cluster partition, where the left-most partition is broken for the first time. In the partitions with larger number of clusters, the results are consistent with those obtained by the scores plots for *Age*. In the 6-cluster partition, cluster 1 contains no patients with very high (huge) *BMI* values, with the majority of the patients having small *BMI* values. On the other hand, cluster 6 contains only patients with large or huge *BMI* values. Thus, as with *Age*, there is clearly discrimination between the four categories of *BMI* in the derived partitions, and the

clustering method works in this case.

To investigate whether the medians of the populations represented by the derived clusters are all equal in the five partitions (whether the *Age* and/or the *BMI* values of the patients play any role in the derivation of the clusters) or not, the KW test will be used with the raw *Age* and *BMI* values of the patients. The *p*-values for the five partitions can be seen in Table 7.14. In all cases, the *p*-values are smaller than the

**Table 7.14:** Kruskal-Wallis rank sum test for the 5 partitions with respect to *Age* and *BMI* to test the equality of the medians of all clusters in each partition. The statistically significant *p*-values at 95% confidence level are shown in bold.

Clusters	P-value	
	Age	BMI
2	<b>0.0034</b>	<b>0.0023</b>
3	<b>0.0109</b>	<b>0.0070</b>
4	<b>0.0063</b>	<b>0.0003</b>
5	<b>0.0151</b>	<b>0.0006</b>
6	<b>0.0305</b>	<b>0.0013</b>

significance level of 0.05, therefore there is enough evidence to reject the null hypotheses of the tests that the clusters in each of these partitions represent populations with equal median values. That is, there is at least one cluster for which, the median value of the population it represents is different than that of the represented populations of the rest of the clusters.

Concerning *Age*, from Table 7.10, the first cluster in the 2-cluster partition contains much younger patients than the second cluster. In the case of *BMI*, in Table 7.12, the first cluster in the 2-cluster partition contains mainly patients with small or medium BMI values, whereas the second cluster is mainly of patients with large to huge BMI values. These facts are even more pronounced for both clinical characteristics in the rest of the partitions with the distributions of the clusters with respect to *Age* differing significantly. Low-numbered clusters contain younger patients than high-numbered clusters, and with respect to *BMI*, low-numbered clusters contain in general, patients with lower BMI values than the high-numbered clusters. It is clear that the clustering method works with respect to the *Age* and *BMI* characteristics.

Comparing the test results (and the clusters of the partitions in the two contingency tables), it seems that there is a pattern with respect to *Age* and *BMI*. More specifically, low-numbered clusters contain younger patients with low BMI values, whereas the high-numbered clusters contain older patients with higher BMI. However, despite the clustering method being capable of discriminating patients with regards to *Age* and *BMI*, and in some cases *Gender*, it was not capable of discriminating the patients with respect to their *Seizure Type* and *Response to AEDs* with a low misclassification rate.

The dominating characteristics that were observed in each of the two groups are summarised below:

1. Males in the (16-26] *Age* category with LRE *Seizure Type*, responders to AEDs and *BMI* values small to medium.
2. Males in the (26-47] and (47-99] *Age* categories with LRE *Seizure Type*, balanced *Response to AEDs* and *BMI* values large to huge.

## 7.6 Partitioning Methods

### 7.6.1 Introduction

This is a popular category of clustering algorithms, based on the optimization of a cost function with the aid of various numerical algorithms. A cost function is given in terms of the input vectors  $x_i$  ( $i = 1, \dots, N_s$ , where  $N_s$  is the number of samples in the data set) and an unknown parameter vector  $\vartheta$ . For the optimum solution to the clustering problem, this parameter has to be estimated such that the derived partition represents as closely as possible the clusters which describe the input vectors  $x_i$ . The shape of clusters determines the type of parameter  $\vartheta$  to be used in the analysis. For example,  $\vartheta$  can be considered as a set of point representatives (e.g. centroids, medoids), a quadric surface or a hyperplane if the clusters are compact, quadric-shaped (e.g hyper-ellipsoids or hyperparaboloids) or hyperplanar respectively (Theodoridis and Koutroumbas, 2003).

Optimal partitioning methods are non-hierarchical clustering algorithms which split the objects to be clustered into a predefined number of clusters, say  $N_c$ , such that there is no hierarchical relationship between the  $N_c$  and the  $N_c + 1$  solution (Izenman, 2008). The objects are partitioned into  $N_c$  clusters such that the items in one cluster are similar to each other but different from those in other clusters. These methods are iterative and use only a limited amount of enumeration. Therefore, as they do not need to store large proximity matrices, they are computationally more efficient than hierarchical methods. Such algorithms include, among others, *Sequential*, *Probabilistic*, *Possibilistic*, *Hard* and *Fuzzy* clustering algorithms (Theodoridis and Koutroumbas, 2003).

### 7.6.2 Fuzzy Clustering Algorithms

#### 7.6.2.1 Introduction

These algorithms are based on the concept that an input vector  $x_i$  can belong to a certain degree to more than one cluster at the same time. They are independent of the shape of clusters, so they are efficient in obtaining an optimal partition in any case (Theodoridis and Koutroumbas, 2003). The degree of membership of a vector  $x_i$  to a



cluster  $c$  is determined by a membership coefficient,  $m_{ic}$ . A fuzzy  $m$ -clustering of a matrix  $X$  (containing all the input vectors  $x_i$ ) can then be defined as a set of functions

$$m_c : X \rightarrow A, \quad c = 1, \dots, N_c$$

where  $A = [0, 1]$ . If  $A = \{0, 1\}$  then a hard  $m$ -clustering is defined, such that each input vector belongs to one and only cluster. Considering the case of point representatives, if  $\vartheta \equiv [\vartheta_1^T, \dots, \vartheta_{N_c}^T]^T$ , such that  $\vartheta_i$  is the representative of the  $i^{\text{th}}$  cluster, then a general form of a cost function for a *fuzzy c-means* clustering algorithm is

$$\sum_{i=1}^{N_s} \sum_{j=1}^{N_c} m_{ij}^q d(x_i, \vartheta_j)$$

with respect to  $\vartheta$  and  $M$ , subject to the constraints

$$\begin{aligned} \sum_{j=1}^{N_c} m_{ij} &= 1, \\ m_{ij} &\in [0, 1], \\ 0 < \sum_{i=1}^{N_s} m_{ij} < N_s, \quad i &= 1, \dots, N_s, \quad j = 1, \dots, N_c \end{aligned}$$

where  $M$  is an  $(N_s \times N_c)$  matrix with elements  $(i, j) = (m_j(x_i), d(x_i, \vartheta_j))$  representing the dissimilarity between input vector  $x_i$  and parameter vector  $\vartheta_j$ , and  $q$  ( $\geq 1$ ) is a parameter called the *fuzzifier*. Assuming  $\vartheta$  is fixed, if the fuzzifier parameter is  $q = 1$  then there is no better fuzzy clustering than the best hard-clustering solution, whereas if  $q > 1$  then it is possible to obtain fuzzy clustering optimal solutions which are better than the best hard clustering solution ([Theodoridis and Koutroumbas, 2003](#)).

Fuzzy clustering algorithms have not been used much in metabo(lo)nomics, but in general, have proved quite useful in the analysis of metabolic profiles. The most popular fuzzy clustering method is the *fuzzy c-means* (FCM). An example of its use, is the clustering of *Escherichia coli* gene types on the basis of their metabolic profiles. The method successfully clustered the samples, revealing main phenotype changes in the metabolic profiles and allowing the identification of significantly changed metabolites ([Li et al., 2009](#)). Other research involves the use of FCM in proton NMR metabolomics samples of cancer cell line extracts and of urine of type 2 diabetes patients and animal methods ([Culf et al., 2009](#)). FCM was able to classify more accurately the samples in both data sets in comparison to other methods such as PCA, HCA and  $k$ -means, by clearly separating the individual cell lines, both groups of cancer and normal cell lines and non-invasive and invasive tumour cell lines. In the case of the diabetes data, only FCM was capable of clearly separating healthy controls and diabetics in all the methods that were used.

An important type of fuzzy c-means clustering algorithm is the **fanny** algorithm, described in Section 7.6.2.2. A statistic to assess the fuzziness of a clustering solution, *Dunn's partition coefficient*, is given in Section 7.6.2.3. The application of the **fanny** clustering algorithm to the epilepsy data can be seen in Section 7.6.3.

### 7.6.2.2 The Fanny Fuzzy Clustering Algorithm

This fuzzy clustering algorithm has been developed by Kaufman and Rousseeuw. In this case, the cost function to be minimized is given by

$$\sum_{c=1}^{N_c} \frac{\sum_{i,j=1}^{N_s} m_{ic}^2 m_{jc}^2 d(x_i, x_j)}{2 \sum_{j=1}^{N_s} m_{jc}^2}$$

subject to the constraints

$$m_{ic} \geq 0 \quad \text{for } i = 1, \dots, N_s \quad c = 1, \dots, N_c \quad (7.6.1)$$

$$\sum_c m_{ic} = 1 \quad \text{for } i = 1, \dots, N_s \quad (7.6.2)$$

where  $d(x_i, x_j)$  is the distance (or dissimilarity) between objects  $x_i$  and  $x_j$ ,  $m_{ic}$  the unknown membership coefficient of object  $i$  to cluster  $c$ ,  $N_s$  the number of samples in the data set and  $N_c$  the number of clusters. The first constraint ensures that no membership coefficient is negative and the second that each object has a constant total membership distributed over the  $c$  clusters.

An advantage of this algorithm is that it uses only inter-object dissimilarities, not involving any averages of objects. This algorithm is also more robust to the assumption of spherical clusters, as the distances in the objective function's formula are not squared. A detailed description of the numerical algorithm used to optimally minimize the above cost function is given in Kaufman and Rousseeuw (2005). It should be noted that when the chosen distance metric is the squared Euclidean distances (the sum of squares of differences), the algorithm becomes the standard *fuzzy c-means* method.

### 7.6.2.3 Fuzziness of a Clustering Solution

The fuzziness of a clustering solution, that is, how close to a hard clustering solution is, can be estimated using *Dunn's partition coefficient*. This statistic can be calculated using the following formula

$$F_c(M) = \sum_{i=1}^{N_s} \sum_{c=1}^{N_c} \frac{m_{ic}^2}{N_s}$$

where  $M$  is the matrix of membership coefficients  $m_{ic}$  (Kaufman and Rousseeuw, 2005). The value of this statistic is the sum of squares of all membership coefficients divided by the number of objects. In a hard clustering solution, the statistic obtains its maximum value, 1, whereas it takes its minimum value,  $\frac{1}{N_c}$ , when all membership coefficients have the value  $\frac{1}{N_c}$ . Thus, the statistic's values are in the range  $[\frac{1}{N_c}, 1]$ . The normalized version of the statistic

$$F'_c(M) = \frac{F_c(M) - (1/N_c)}{1 - (1/N_c)} = \frac{N_c F_c(M) - 1}{N_c - 1}$$

is more straightforward as it takes values in  $[0, 1]$ , where 0 means total fuzziness and 1 a completely hard solution.

## 7.6.3 Application of Fuzzy Clustering to the Epilepsy Data

### 7.6.3.1 Introduction

The data that will be analysed by fuzzy clustering is the same that was used in the hierarchical clustering analyses. The data set includes the 97 patients with specific *response to AEDs* information (only responders or non-responders), with intensity values in the proton NMR chemical shift range of 5.98 – 0.02 *ppm*. The data has also been row-scaled to a constant total.

The **fanny** fuzzy clustering algorithm will be used to analyse the data, as described in Section 7.6.2.2. The reason for this is that according to the authors of the function, and as mentioned in Section 7.6.2.2, **fanny** compared to other fuzzy clustering methods, can accept as data input also a dissimilarity matrix. In addition, it is more robust to the spherical cluster assumption and provides silhouette information (silhouette widths and plot) which can be used to assess the quality of the results obtained by a fuzzy clustering method.

Things to be considered in order to compare the results of the fuzzy clustering analyses are the distance measure, the number of required clusters and the value of the *fuzzifier* (membership exponent). Four different distance metrics will be used to compute the distance matrix of the observations in the epilepsy data, i.e. the *Euclidean*, the *Manhattan*, the *Maximum* and the *SqEuclidean* (sum of squares of differences), as it is not clear which distance metric is the best for this type of data. It should be noted that, according to the authors of the **fanny** function, using the *SqEuclidean* metric results in a fuzzy clustering analysis equivalent to the fuzzy *c*-means method. The number of clusters will be chosen in the range 2-6, and the *fuzzifier* values to be used include 1.1, 1.5, 2, 2.5 and 3.0.

The best (if any) fuzzy clustering method will be identified using tools such as the optimal (minimum) objective function value, *Dunn's partition coefficient*, as well as its

*normalised* version and the *silhouette* information. These will allow the selection of the optimal method with respect to the optimum number of clusters, distance measure and *fuzzifier* value. The *silhouette coefficients* will be used in the optimal method with respect to the distance measure and *fuzzifier* value to determine the optimum number of clusters for the selected method.

The  $\chi^2$ , Fisher's and KW tests will be used to assess whether the distributions of the populations represented by the clusters of the optimal partition with regards to the various clinical characteristics have differences or not, as has already been done in the case of the hierarchical clustering methods.

### 7.6.3.2 Comparison of Fuzzy Clustering Methods

To examine how the value of the *fuzzifier* parameter affects the fuzzy clustering solution, a number of runs were performed using various *fuzzifier* values for fuzzy 2-cluster partitions with the four pre-selected distance metrics. A comparison of various **fanny** fuzzy clustering methods with respect to the distance measure and the value of the *fuzzifier* can be seen in Table 7.15. The table shows that the best overall fuzzy 2-cluster method

**Table 7.15:** Comparison of fuzzy clustering methods with regards to pre-selected *fuzzifier* values and distance measures. The number of clusters is set to 2. OASW stands for the overall *average silhouette width*. In bold are shown the clustering methods which give the best results (for each metric) with respect to the pair of values of OASW and the objective function. The different number of decimals in some of the entries of the table is due to the fact that all values were put in the table as they were returned by the analyses in R (and are not a result of any rounding).

Metric	Fuzzifier	Iterations	Objective	Dunn	Dunn (Norm)	OASW
Euclidean	3.0	40	0.22748	0.51946	0.03892	0.51761
Euclidean	2.5	36	0.31684	0.55087	0.10174	0.51761
Euclidean	2.0	34	0.43003	0.63566	0.27132	0.52752
<b>Euclidean</b>	<b>1.5</b>	<b>31</b>	<b>0.54072</b>	<b>0.83974</b>	<b>0.67948</b>	<b>0.53194</b>
Euclidean	1.1	16	0.57708	0.98449	0.96898	0.53582
Manhattan	3.0	73	1.0955	0.5	7.9e-15	0.42119
Manhattan	2.5	220	1.5491	0.50278	0.00555	0.42119
Manhattan	2.0	30	2.1641	0.55831	0.11662	0.42561
Manhattan	1.5	19	2.8407	0.76302	0.52603	0.44208
<b>Manhattan</b>	<b>1.1</b>	<b>19</b>	<b>3.0913</b>	<b>0.98165</b>	<b>0.96331</b>	<b>0.44984</b>
Maximum	3.0	24	0.16755	0.55111	0.10221	0.58442
Maximum	2.5	21	0.22974	0.59696	0.19392	0.58442
Maximum	2.0	28	0.30492	0.69721	0.39443	0.58761
<b>Maximum</b>	<b>1.5</b>	<b>16</b>	<b>0.37398</b>	<b>0.87935</b>	<b>0.7587</b>	<b>0.59384</b>
Maximum	1.1	29	0.39733	0.97841	0.95682	0.59384
SqEuclidean	3.0	20	0.00444	0.6538	0.3076	0.73667
SqEuclidean	2.5	20	0.00584	0.72398	0.44797	0.73667
<b>SqEuclidean</b>	<b>2.0</b>	<b>31</b>	<b>0.00734</b>	<b>0.8251</b>	<b>0.65019</b>	<b>0.74391</b>
SqEuclidean	1.5	17	0.00854	0.93272	0.86544	0.74391
SqEuclidean	1.1	26	0.00894	0.98437	0.96873	0.74391

with respect to its overall *average silhouette width* of 0.74391, is the *SqEuclidean* with

*fuzzifier* value of 2. Despite the methods with *fuzzifier* values 1.5 and 1.1 having the same value of overall *average silhouette width* as the best method, these methods are far less fuzzy than the former method, as indicated by the values of the *Dunn* statistics. In the case of the *Euclidean* methods, the overall *average silhouette width* values of the methods with *fuzzifier* 1.5 and 1.1 are 0.53194 and 0.53582 respectively. These values are very close, therefore, as this is fuzzy clustering analysis, the former method is chosen as the best fuzzy method for the *Euclidean* metric. The *Manhattan* metric is clearly not suitable for fuzzy clustering analysis, as all the *Manhattan* methods in question have the lowest overall *average silhouette width* values among all metric distances. Due to this fact, the best, and the chosen, *Manhattan* method is the one with *fuzzifier* 1.1 and overall *average silhouette width* value of 0.44984, which if it remains at these levels for all possible numbers of clusters, will mean that the structure of the data is weak. The second best, among all metrics, overall *average silhouette width* values can be found in the *Maximum* methods, which seems to be consistent with the results obtained from HCA (although the metric *SqEuclidean* was not used in the HCA methods). The *Maximum* method with *fuzzifier* 1.5 and overall *average silhouette width* value of 0.59384 is the best among the *Maximum* methods, and is retained for further analyses.

### 7.6.3.3 Identification of the Optimal Number of Clusters

The four retained methods among all metrics and *fuzzifier* values, as seen in Table 7.15, will be analysed further for the cases of partitions having 2-6 clusters, with the purpose of identifying the optimal number of clusters. The reason for not retaining for further analyses only the best clustering method with respect to the overall *average silhouette width* value, is the need to confirm whether or not the number of clusters plays any role in the value of the overall *average silhouette width*. A comparison of the analyses of the four clustering methods with partitions of 2-6 clusters can be seen in Table 7.16. The *fuzzifier* parameter has been set equal to 1.5, 1.1, 1.5 and 2.0 for the methods *Euclidean*, *Manhattan*, *Maximum* and *SqEuclidean* respectively in all partitions in Table 7.16. As is logical, *Dunn's coefficient* (and its normalised version) reduces as the number of clusters increases. That is, the larger the number of clusters in the derived partition, the lower the value of *Dunn's coefficient* is (the greater the fuzziness of the clustering solution is). From Table 7.16, it is clear that the best overall fuzzy clustering method is derived for two clusters using the *SqEuclidean* distance as the metric to calculate the dissimilarities among the observations, as its overall *average silhouette value*, 0.74391, is the highest among all clustering methods in the table. The data is well-structured for this partition. In general, fuzzy 2-cluster partitions seem to fit the data better, independently of the metric used. The *SqEuclidean* metric is the most efficient in fitting the data, as even when a fuzzy 6-cluster partition is selected, the structure of the data is close to being reasonable with its overall *average silhouette width* value being 0.48742.

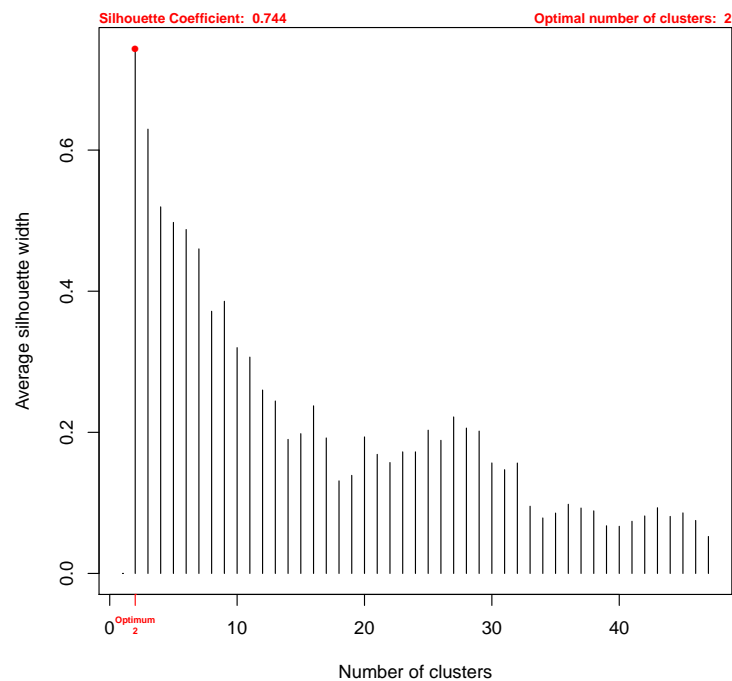
**Table 7.16:** Comparison of fuzzy clustering methods with regards to pre-selected number of clusters and distance measures. The *fuzzifier* values are the same as those of the best methods for the four distance metrics in Table 7.15. OASW stands for the overall *average silhouette width*. In bold is shown the fuzzy clustering method which gives the best result with respect to the value of OASW and if necessary, the objective function. The different number of decimals in some of the entries of the table is due to fact that all values were put in the table as they were returned by the analyses in R (and are not a result of any rounding).

Clusters	Metric	Iterations	Objective	Dunn	Dunn (Norm)	OASW
2	Euclidean	31	0.54072	0.83974	0.67948	0.53194
	Manhattan	19	3.0913	0.98165	0.96331	0.44984
	Maximum	16	0.37398	0.87935	0.7587	0.59384
	<b>SqEuclidean</b>	<b>31</b>	<b>0.00734</b>	<b>0.8251</b>	<b>0.65019</b>	<b>0.74391</b>
3	Euclidean	277	0.4309	0.6485	0.47275	0.33149
	Manhattan	70	2.7191	0.94597	0.91895	0.2925
	Maximum	32	0.28536	0.80841	0.71262	0.51676
	SqEuclidean	57	0.00439	0.69228	0.53843	0.62966
4	Euclidean	30	0.35567	0.60939	0.47918	0.34561
	Manhattan	26	2.43	0.94607	0.9281	0.2677
	Maximum	23	0.22846	0.72013	0.62683	0.43039
	SqEuclidean	65	0.00295	0.59456	0.45941	0.51954
5	Euclidean	61	0.31411	0.51477	0.39346	0.31654
	Manhattan	37	2.2627	0.93848	0.9231	0.26332
	Maximum	216	0.20212	0.62337	0.52921	0.35796
	SqEuclidean	150	0.00226	0.55099	0.43874	0.49749
6	Euclidean	112	0.28299	0.49266	0.39119	0.30385
	Manhattan	118	2.1346	0.93108	0.9173	0.23864
	Maximum	98	0.17614	0.60691	0.5283	0.35626
	SqEuclidean	54	0.00177	0.48979	0.38775	0.48742

Figure 7.16 gives the overall *average silhouette widths* of the best fuzzy clustering method of Table 7.16 with the *SqEuclidean* metric and *fuzzifier* value of 2, for all partitions from 2-47 clusters. The maximum number of clusters in a *fanny* partition is  $\frac{n}{2} - 1$ , therefore, as in the epilepsy data  $n$  is equal to 97, the maximum number of clusters is 47. The degree of membership (*membership coefficients*) of the 97 patients to the 2 clusters in the optimum *fanny* partition can be seen in Table 7.17. The membership values for patients 13, 18, 56 and 92 are quite balanced for the two clusters meaning that the chosen clustering method has not been able to categorize these patients to either of the two clusters successfully. Three other patients, namely 28, 50 and 110, have *membership coefficients* indicating one or the other cluster but these are rather moderate (0.60-0.65), therefore they do not belong very strongly to their respective clusters. However, the majority of the patients have *membership coefficients* above 0.85, so the selected fuzzy clustering method seems to cluster well the data. It remains to be seen whether the selected fuzzy clustering method can discriminate the patients with respect to their clinical characteristics.

**Table 7.17:** Membership coefficients for the fuzzy 2-cluster partition obtained by the fanny method with the *SqEuclidean* metric and *fuzzifier* value 2. In bold are shown the largest of the two *membership coefficients* for each of the patients.

Patient	Cluster 1	Cluster 2	Patient	Cluster 1	Cluster 2
001	<b>0.89789</b>	0.10210	063	0.28798	<b>0.71201</b>
002	<b>0.97210</b>	0.02789	064	0.27254	<b>0.72745</b>
003	<b>0.95434</b>	0.04565	065	<b>0.72238</b>	0.27761
004	<b>0.90067</b>	0.09932	066	0.06864	<b>0.93135</b>
005	<b>0.98077</b>	0.01922	068	0.27908	<b>0.72091</b>
006	<b>0.98041</b>	0.01958	069	<b>0.99069</b>	0.00930
007	<b>0.73638</b>	0.26361	070	<b>0.94026</b>	0.05973
008	0.05531	<b>0.94468</b>	074	<b>0.97878</b>	0.02121
009	<b>0.83720</b>	0.16279	075	<b>0.92017</b>	0.07982
010	0.03100	<b>0.96899</b>	076	<b>0.95796</b>	0.04203
011	<b>0.94942</b>	0.05057	077	<b>0.98310</b>	0.01689
012	<b>0.66878</b>	0.33121	078	<b>0.63952</b>	0.36047
013	<b>0.54106</b>	0.45893	079	<b>0.87566</b>	0.12433
014	0.09275	<b>0.90724</b>	080	<b>0.89758</b>	0.10241
015	<b>0.88204</b>	0.11795	081	<b>0.97353</b>	0.02646
016	<b>0.96350</b>	0.03649	082	0.03815	<b>0.96184</b>
017	<b>0.94671</b>	0.05328	087	<b>0.79947</b>	0.20052
018	0.46290	<b>0.53709</b>	089	<b>0.97547</b>	0.02452
019	0.10710	<b>0.89289</b>	090	<b>0.98781</b>	0.01218
020	0.21147	<b>0.78852</b>	091	<b>0.88940</b>	0.11059
021	<b>0.93419</b>	0.06580	092	<b>0.50292</b>	0.49707
024	<b>0.96301</b>	0.03698	093	<b>0.95184</b>	0.04815
028	0.39543	<b>0.60456</b>	096	<b>0.93462</b>	0.06537
029	<b>0.92234</b>	0.07765	097	0.01003	<b>0.98996</b>
030	<b>0.93072</b>	0.06927	098	0.12414	<b>0.87585</b>
032	<b>0.93897</b>	0.06102	099	0.21114	<b>0.78885</b>
033	0.12501	<b>0.87498</b>	100	<b>0.92044</b>	0.07955
035	0.04287	<b>0.95712</b>	101	0.17739	<b>0.82260</b>
036	0.14191	<b>0.85808</b>	104	<b>0.79656</b>	0.20343
037	<b>0.93919</b>	0.06080	105	<b>0.96778</b>	0.03221
041	<b>0.94619</b>	0.05380	106	<b>0.98667</b>	0.01332
042	0.04400	<b>0.95599</b>	107	<b>0.95408</b>	0.04591
044	<b>0.89802</b>	0.10197	109	<b>0.93806</b>	0.06193
045	<b>0.94295</b>	0.05704	110	0.34845	<b>0.65154</b>
046	0.02514	<b>0.97485</b>	111	<b>0.94266</b>	0.05733
048	<b>0.87648</b>	0.12351	112	<b>0.82025</b>	0.17974
049	0.11305	<b>0.88694</b>	113	<b>0.99284</b>	0.00715
050	0.34574	<b>0.65425</b>	114	0.06378	<b>0.93621</b>
051	<b>0.92179</b>	0.07820	115	<b>0.95156</b>	0.04843
052	0.01344	<b>0.98655</b>	116	0.11247	<b>0.88752</b>
053	<b>0.88286</b>	0.11713	117	<b>0.94726</b>	0.05273
054	<b>0.97644</b>	0.02355	118	<b>0.97669</b>	0.02330
055	<b>0.99254</b>	0.00745	119	<b>0.95935</b>	0.04064
056	<b>0.52016</b>	0.47983	120	<b>0.94673</b>	0.05326
058	<b>0.93060</b>	0.06939	122	<b>0.09088</b>	0.90911
059	0.17163	<b>0.82836</b>	123	0.04991	<b>0.95008</b>
060	<b>0.93917</b>	0.06082	124	0.03576	<b>0.96423</b>
061	<b>0.93249</b>	0.06750	125	<b>0.94084</b>	0.05915
062	<b>0.93763</b>	0.06236			

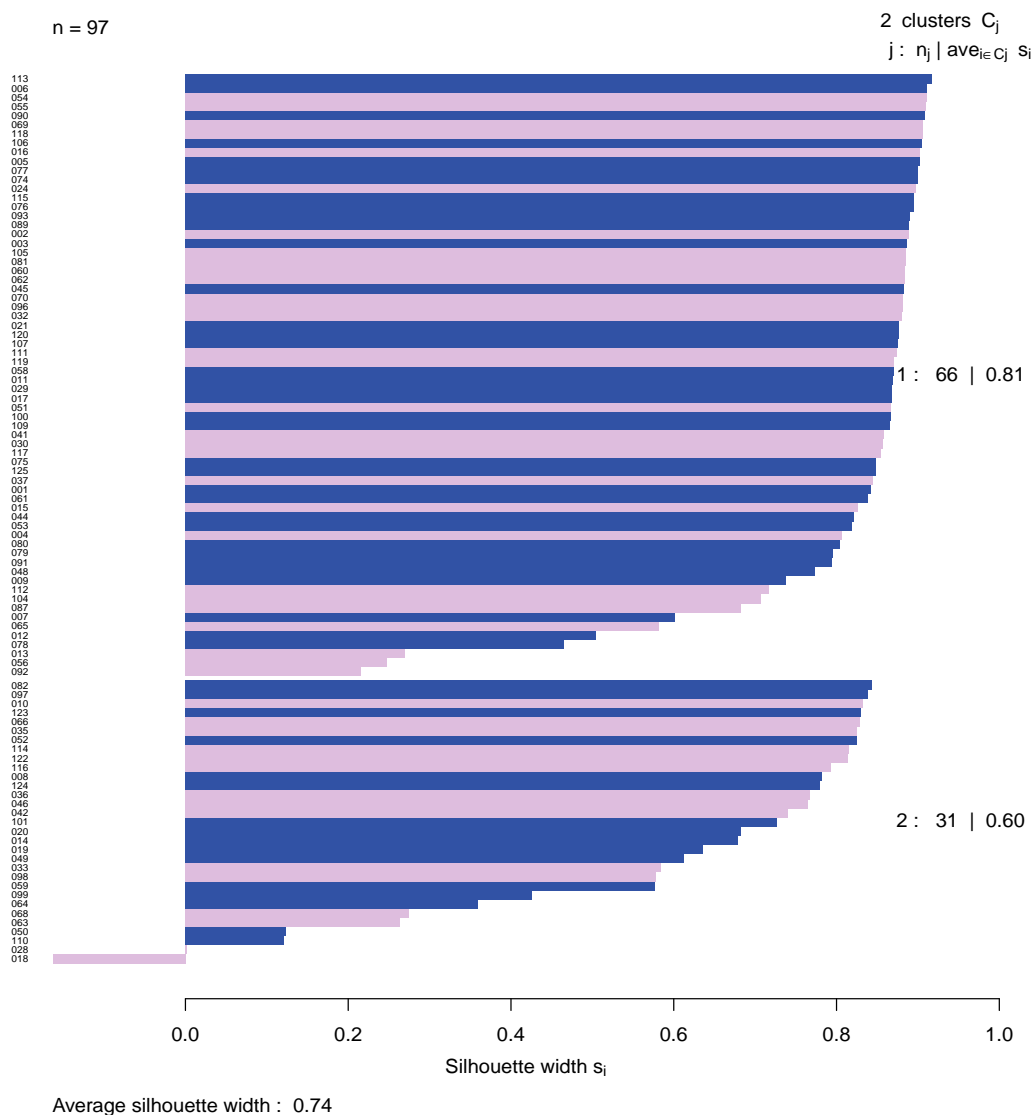


**Figure 7.16:** Average silhouette widths for partitions of 2-47 clusters for the selected fuzzy clustering method (*SqEuclidean* metric and *fuzzifier* value 2). The optimal number of clusters is indicated in red.

#### 7.6.3.4 Discrimination of the Clinical Characteristics

The capability of the selected clustering method to discriminate the patients with respect to their clinical characteristics can be assessed in many ways. A *silhouette plot* can confirm the findings for the *membership coefficients* and the *response to AEDs* information of the patients. The *silhouette plot* for the selected fuzzy clustering method can be seen in Figure 7.17. The silhouette values confirm the findings concerning the membership coefficients for patients 13, 18, 56 and 92. More specifically, patient 18 has negative *silhouette width* and has been wrongly classified to cluster 2 instead of 1. Patients 28, 50 and 110 have very small *silhouette width* values around 0.1, so they have been classified poorly in cluster 2. In addition, patients 13, 56 and 92, which have balanced *membership coefficients*, as discussed previously, have been weakly classified to cluster 1 with very small *silhouette width* values (around 0.2). Therefore, the *silhouette plot* indicates clearly, that the *silhouette widths* (depicted as bars in the plot) for all these patients are consistent with their *membership coefficients*. The *silhouette plot* shows that the first and second cluster contain 66 and 31 patients respectively. Compared to the results from the best HCA method, this fuzzy clustering method fits the epilepsy data better, as the overall *average silhouette width* of 0.74 for the fuzzy method is clearly higher than that of the HCA method (0.59). The *average silhouette*





**Figure 7.17:** *Silhouette plot* for the 2-cluster partition derived by the *SqEuclidean - fuzzifier 2* fuzzy clustering method. The blue and the pink bars correspond to responders and non-responders respectively. The *average silhouette width* for clusters 1 and 2 is 0.81 and 0.60 respectively, and the *average silhouette width* for the entire data set is 0.74.

*width* for the two clusters of the fuzzy partition are also higher than those of the HCA partition, being 0.81 and 0.60 for clusters 1 and 2 of the fuzzy partition respectively, while for the HCA partition they are 0.68 and 0.43 for cluster 1 and 2 respectively. In addition, the plot confirms the results of Table 7.18, indicating clearly that there is no discrimination of the patients, with low misclassification rate, with respect to their *Response to AEDs*.

Table 7.18 gives the results of cross-tabulating the *response to AEDs* information with the clusters of the fuzzy method. The results of the table show that the misclassification rate of *Response* in the fuzzy clustering method is not smaller than that of the optimal

**Table 7.18:** Cross-tabulation of the optimal 2-cluster fuzzy partition to *Response to AEDs*.

Clusters	Response to AEDs	
	Responder	Non-responder
1	36	30
2	16	15

HCA method, as there are  $16 + 30 = 46$  misclassified patients when using the fuzzy method. This result is consistent with the HCA findings. Graphical tools can also confirm that the fuzzy clustering method cannot discriminate with low misclassification rates between responders and non-responders to AEDs.

To investigate the homogeneity of the 2-cluster partition in Table 7.18 with respect to the observations in each cell for the two categories of *Response to AEDs*, the  $\chi^2$  test will be used. The  $p$ -value of the test, 0.9587, is larger than the significance level of 0.05, therefore there is not enough evidence to reject the null hypothesis of the test that the clusters in the partition are homogeneous with respect to the *Response to AEDs*. The proportions of observations in each of the categories of *Response to AEDs* are not different in the selected partition.

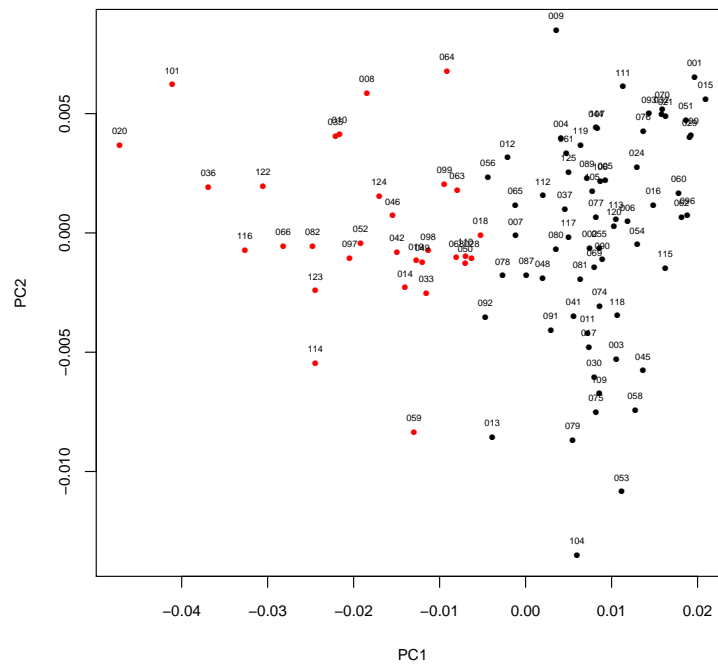
To illustrate the clustering solution derived by the *SqEuclidean - Fuzzifier* value 2 fuzzy clustering method, a two-dimensional projection of the clustering solution can be seen in Figure 7.18. The first two principal component scores (according to the results from Chapter 5) can be seen superimposed with the partition derived by the selected fuzzy clustering method. In both scores plots, black and red represent the patients clustered to the first and second cluster respectively. The bottom scores plot illustrates the *Response to AEDs* information, such that points labelled as "R" and "N" correspond to responders or non-responders to AEDs respectively. It is clear from Figure 7.18 that the derived clusters are not compact. Also, there is no distinction among responders and non-responders to AEDs by this clustering. This algorithm has not been efficient in classifying patients according to their *response to AEDs*.

Table 7.19 contains the contingency tables between the selected 2-cluster fuzzy partition and the clinical characteristics *Gender* and *Age*.

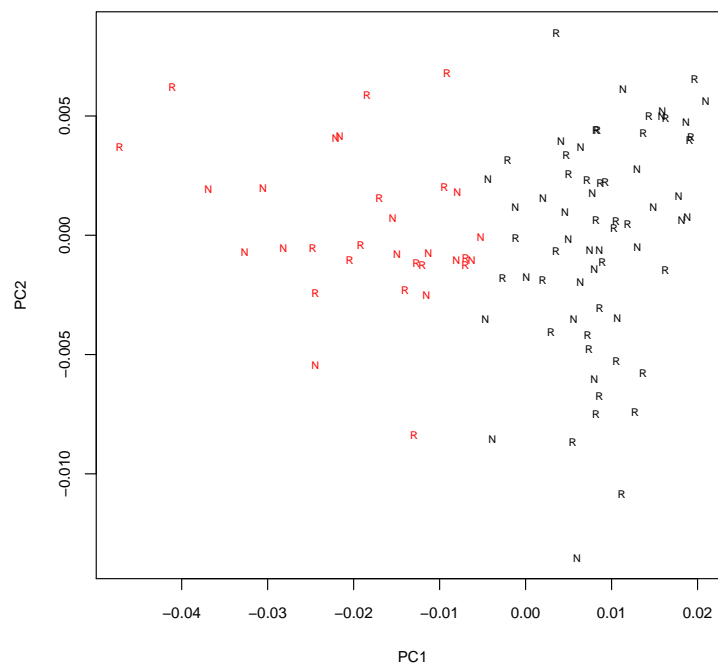
**Table 7.19:** Cross-tabulation of the optimal 2-cluster fuzzy partition to *Gender* and *Age* clinical characteristics.

Cluster	Gender		Age		
	Male	Female	(16-26]	(26-47]	(47-99]
1	43	23	28	20	18
2	22	9	3	12	16

The cross-tabulation of *Gender* shows that, as expected due to their large number in the data set, males dominate both clusters in the partition. This is even more



(a) 2-cluster partition



(b) Response to AEDs

**Figure 7.18:** Scores plots of the first two PCs, superimposed with the 2-cluster partition derived by the *SqEuclidean - Fuzzifier 2* fuzzy clustering method. Black and red represent the patients in the first and second cluster respectively. The labels of the points in the bottom plot correspond to the responders (R) and non-responders (N) to AEDs.

pronounced due to the fact that the first cluster contains approximately twice as many patients as the second. Concerning the *Age* of the patients, as in the optimal HCA method, the first cluster, although rather balanced in all *Age* categories, contains more young patients than the other two categories. However, the second cluster is clearly dominated by middle-aged and old patients, having only 3 young patients among the 29 patients of cluster 2. There seems to be a pattern with respect to *Age*.

The  $p$ -value of the  $\chi^2$  test for homogeneity of the 2 clusters with respect to the *Gender* is 0.7364. There is not sufficient evidence to reject the null hypothesis that the 2 clusters are homogeneous with regards to *Gender*, therefore the proportions of patients in the two *Gender* categories are not different in the selected partition.

Concerning the contingency table of *Age*, results are similar to those of the 2-cluster HCA partition. That is, the first cluster in the fuzzy partition contains more patients of younger ages (most being young) in a ratio of approximately 3:2 (young compared to each of the other two categories), whereas the second cluster contains only 3 young patients with the patients of the other two categories dominating the cluster.

The  $p$ -value of the KW test for the select fuzzy clustering partition is 0.001741, which is much smaller than the significance level of 0.05, therefore the null hypothesis is rejected and the populations represented by the two clusters in the partition have different median values. There is clearly a relationship between the clusters and the *Age* of the patients.

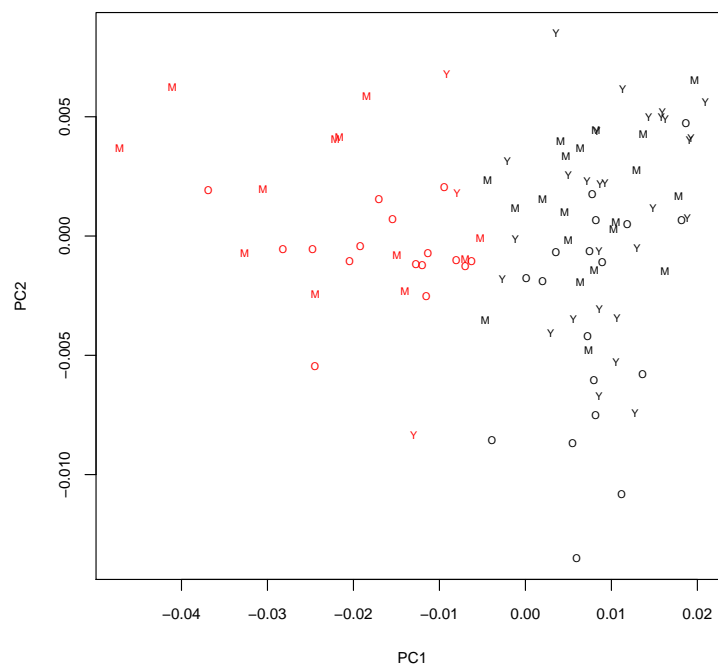
The selected fuzzy clustering method is depicted as a two-dimensional projection of the data superimposed with the *Age* information in Figure 7.19. The first two principal component scores (according to the results from Chapter 5) can be seen superimposed with the partition derived by the selected fuzzy clustering method and the points of the plot labelled as young (Y), middle-aged (M) and old (O) patients. Black and red points represent the patients clustered to the first and second cluster respectively. The relationship of the clusters with the *Age* is clear, as the second cluster contains only 3 young patients, whereas the majority of patients in the first cluster are young. Thus, the selected fuzzy clustering method works in this case, as it was also shown by the results of the KW test.

The contingency tables for *Seizure Type* and *BMI* can be seen in Table 7.20. The

**Table 7.20:** Cross-tabulation of the optimal 2-cluster fuzzy partition to the *Seizure Type* and *BMI* clinical characteristics.

Cluster	Seizure Type		BMI			
	LRE	IGE	(16-22]	(22-25]	(25-28]	(28-45.1]
1	47	19	20	21	14	11
2	28	3	5	6	8	12

results of Table 7.20 for the *Seizure Type* show a similarity to those of the best HCA



**Figure 7.19:** Scores plots of the first two PCs, superimposed with the 2-cluster fanny partition, derived by the *SqEuclidean - Fuzzifier 2* clustering method and the *Age* information. The labels of the points in the plots correspond to the young (Y), middle-aged (M) and old (O) patients. Black and red points correspond to the first and second cluster respectively.

method. That is, both clusters are dominated by LRE patients with ratios 5:2 and 6:1 in cluster 1 and 2 respectively. That is expected as the number of LRE patients in the data set is considerably larger than that of the IGE patients.

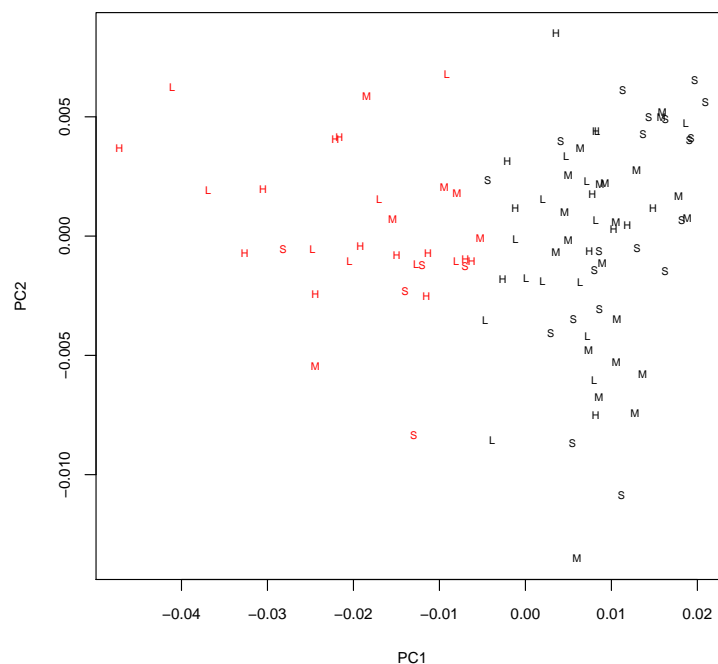
The  $p$ -value of the  $\chi^2$  test for the homogeneity of the 2 clusters with respect to the *Seizure Type* is 0.06637, which is larger than the significance level of 0.05. Hence, there is not sufficient evidence to reject the null hypothesis that the 2 clusters are homogeneous with regards to *Seizure Type*, therefore the proportions of patients with respect to the two *Seizure Type* categories are not significantly different in the two clusters of the selected partition.

Concerning *BMI*, it is clear that there is a discrimination of the patients, as the first cluster contains mainly patients with small and medium *BMI* values (with a ratio of approximately 2:1 to the patients with large or huge *BMI* values), whereas the second cluster is dominated by patients with large and huge *BMI* values with similar ratio to the patients having small or medium *BMI* values.

The  $p$ -value of the KW test for the select fuzzy clustering partition is 0.01079, which is much smaller than the significance level of 0.05, therefore the null hypothesis is rejected and the populations represented by the two clusters in the partition, have different

median values. There is clearly a relationship between the clusters and the *BMI* values of the patients. Thus, the selected fuzzy clustering method manages to discriminate the patients with respect to *BMI*.

A two-dimensional projection of the data, superimposed with the fuzzy clustering result and the *BMI* information can be seen in Figure 7.20. The scores plot for *BMI*



**Figure 7.20:** Scores plots of the first two PCs, superimposed with the 2-cluster *fanny* partition, derived by the *SqEuclidean - Fuzzifier 2* clustering method and the *BMI* information. The labels of the points in the plots correspond to patients with small (S), medium (M), large (L) and huge (H) *BMI* values. Black and red correspond to the first and second cluster respectively.

shows that indeed there are many more patients with low and medium *BMI* values than large and huge values in the first cluster (black symbols), whereas in the second cluster (red symbols) the majority of patients have large or huge *BMI* values.

Comparing the optimal 2-cluster partitions derived by HCA and fuzzy clustering, it can be seen that concerning the patients in the two clusters of the partitions, the only difference is that in cluster 1 of the fuzzy clustering partition there are three patients, namely 13, 78 and 92, who have been assigned to cluster 2 in the HCA clustering partition. The dominating characteristics that were observed in each of the two groups of the fuzzy clustering method are quite similar to those of the HCA method, as the difference of only three patients does not change significantly the dominating characteristics of the patients in the two clusters. Therefore, similarly to the HCA clustering method results, the dominating characteristics of the patients in the two clusters are summarised as:

1. Males in the (16-26] *Age* category with LRE *Seizure Type*, responders to AEDs and *BMI* values small to medium.
2. Males in the (26-47] and (47-99] *Age* categories with LRE *Seizure Type*, balanced *Response to AEDs* and *BMI* values large to huge.

## 7.6.4 Hard Clustering Algorithms

### 7.6.4.1 Introduction

In this case, each input vector  $x_i$  belongs exclusively to one and only cluster. A hard  $m$ -clustering of a data set of a matrix  $X$  (containing all the input vectors  $x_i$ ) can then be defined as a set of functions

$$m_c : X \rightarrow A, \quad c = 1, \dots, N_c$$

where  $A = \{0, 1\}$ . A common form of a cost function for a hard clustering algorithm is given by

$$\sum_{i=1}^{N_s} \sum_{j=1}^{N_c} m_{ij} d(x_i, \vartheta_j) \quad (7.6.3)$$

subject to the constraints

$$m_{ij} \in \{0, 1\}, \quad i = 1, \dots, N_s, \quad j = 1, \dots, N_c$$

$$\sum_{j=1}^{N_c} m_{ij} = 1,$$

where  $M$  is as in the case of fuzzy clustering. Equation 7.6.3 is minimized when each input vector  $x_i$  is assigned to its closest cluster

$$m_{ij} = \begin{cases} 1, & \text{if } d(x_i, \vartheta_j) = \min_{1, \dots, N_c} d(x_i, \vartheta_c) \\ 0, & \text{otherwise} \end{cases}$$

as only one  $m_{ij}$  is equal to 1 and all other membership coefficients are equal to 0 for each input vector  $x_i$  (Theodoridis and Koutroumbas, 2003).

The most popular hard clustering algorithm,  $k$ -means, is described in detail in Section 7.6.4.2. This algorithm is applied to the epilepsy data and the results can be seen in Section 7.6.5.

### 7.6.4.2 The $k$ -means Clustering Algorithm

This algorithm is one of the most popular hard clustering algorithms. In this case, point representatives are used (centroids,  $\vartheta$ ) and the squared Euclidean distance used to measure the distance between the input vectors  $x_i$  and the centroids  $\vartheta_j$ . As in this case  $\vartheta_j$  is the mean vector of cluster  $j$ , the derived clusters are as compact as possible. The algorithm is described below (Izenman, 2008; Theodoridis and Koutroumbas, 2003):

1. Given a set of objects  $x_i, i = 1, 2, \dots, n$  and  $N_c$  the number of clusters, initialize the algorithm by one of the following:
  - Randomly assign the objects into  $N_c$  clusters and for each cluster  $c$  compute its current centroid,  $\bar{x}_c$ .
  - Pre-specify  $N_c$  cluster centroids,  $\bar{x}_c, c = 1, 2, \dots, N_c$ .
2. Compute the optimum criterion (here the squared Euclidean distance) of each object to its current centroid

$$ESS = \sum_{c=1}^{N_c} \sum_{c_i=c} (x_i - \bar{x}_c)^T (x_i - \bar{x}_c)$$

where  $\bar{x}_c$  is the  $c$ th cluster centroid and  $c_i$  is the cluster containing  $x_i$ .

3. Re-assign each object to its nearest cluster centroid, such that  $ESS$  is reduced in magnitude. Update the cluster centroids after each reassignment.
4. Repeat steps 3 and 4 until no further reassignment of items takes place.

Advantages of using  $k$ -means clustering are:

- If the number of variables is large, as in the case of metabonomics data, this clustering method can be computationally faster than HCA.
- Due to the fact that the algorithm seeks to minimise the within-clusters sum of squares and maximise the between-clusters sum of squares, this method is likely to produce tighter clusters than HCA.

Despite  $k$ -means being faster and capable of handling greater numbers of observations than hierarchical clustering, there are a number of disadvantages when using it (Myatt, 2007):

- The number of clusters must be defined before creating the clusters
- Outliers can affect the quality of an optimal clustering
- No hierarchical organization is generated using  $k$ -means clustering.
- This technique is more suitable for identifying compact spherical clusters, so it is not the ideal method for clustering data if the shapes of the clusters are not expected to be multivariate normal.
- Different initial partitions can result in different final clusters.



The  $k$ -means algorithm is very popular in many scientific areas, but has not been used widely in metabolic profiling applications. As there are no specific visualisation or diagnostic tools associated with  $k$ -means, it is usually used in combination with other clustering and visualisation methods. One such case is the statistical analysis of NMR spectra from partially purified marine and plant extracts, using three different unsupervised methods, PCA,  $k$ -means and MDS (Pierens et al., 2005).

The algorithm given by Hartigan and Wong (1979) will be applied to the epilepsy data. This is an efficient version of the  $k$ -means algorithm described in detail in Hartigan (1975). Algorithm AS 136 as it is called, is the preferred and default version of  $k$ -means implemented in R, since the majority of authors of  $k$ -means R functions consider that in general it does a better job than other  $k$ -means algorithms such as those developed by MacQueen (MacQueen, 1967) and Lloyd (Lloyd, 1982). Hartigan's AS 136 algorithm contains two main stages, the *optimal-transfer* (OPTRA) and the *quick-transfer* (QTRAN) stage, to search for a  $k$ -cluster partition with locally optimal (minimum) within-cluster sum of squares, by moving objects from one cluster to another. To improve the chance of finding the global minimum, after assigning the objects at random to the various clusters and finding a local optimal solution, the whole procedure is repeated for a pre-specified number of times (usually 100) starting in every run from a different random configuration. The global optimal solution is the one with the minimum within-cluster sum of squares among all runs (Legendre and Legendre, 1998).

The main steps of algorithm AS 136 are given below (Hartigan and Wong, 1979).

**INPUT:** A matrix  $\mathbf{X}$  of  $x_i$ ,  $i = 1, \dots, N_s$  objects in a  $p$ -dimensional space and an initial centroid configuration. This can be either the number of required clusters, say  $N_c$ , or a set of initial centroids for the required number of clusters. In the former case, a random set of distinct rows in  $\mathbf{X}$  is chosen as the initial centroids, say  $\bar{C}_j$ ,  $j = 1, \dots, N_c$ . The number of points in a cluster, say  $c$ , is denoted by  $NS_c$ , and the Euclidean distance between object  $x_i$  and cluster centroid  $\bar{C}_j$  by  $d(x_i, \bar{C}_j)$ .

**Step 1.** For each object  $x_i$ ,  $i = 1, \dots, N_s$ , find its closest and second closest centroids,  $\bar{C}_{1_i}$  and  $\bar{C}_{2_i}$  respectively. Assign object  $x_i$  to cluster  $C_1$ .

**Step 2.** Update the centroids to be the averages of the points that they contain.

**Step 3.** Initially, all clusters are members of the live set.

**Step 4 - OPTRA.** For each object  $x_i$  in turn, if cluster  $C$  is updated in the last QTRAN stage, then it belongs to the live set throughout this stage. Otherwise, at each step, it is not in the live set if it has not been updated in the last  $N_s$  optimal-transfer steps. Let object  $x_i$  be in cluster  $C_j$ . If  $C_j$  is in the live stage, go to **Step 4a**, otherwise go to **Step 4b**.

**Step 4a.** Compute the minimum of the quantity

$$R2 = \frac{NS_c d(x_i, C)^2}{NS_c + 1}$$

for all clusters  $C$  ( $C \neq C_j$ ,  $j = 1, \dots, N_c$ ). Let  $C_l$  be the cluster with the smallest  $R2$ . If this value is greater than or equal to

$$\frac{NS_{C_j} d(x_i, C_j)^2}{NS_{C_j} - 1},$$

no reassignment is necessary and  $C_l$  is the new  $\bar{C}2_i$ . Otherwise object  $x_i$  is assigned to cluster  $C_l$  and  $C_j$  is the new  $\bar{C}2_i$ . Centroids are updated to be means of objects assigned to them if reassignment has taken place. The two clusters that are involved in the transfer of object  $x_i$  at this particular step are now in the live set.

**Step 4b.** The same as Step 4a, with the only exception being that the minimum  $R2$  is computed only for clusters in the live set.

**Step 5.** Stop if the live set is empty. Otherwise go to **step 6** after one pass through the data set.

**Step 6 - QTRAN.** For each object  $x_i$  in turn, let  $C_j = \bar{C}1_i$  and  $C_l = \bar{C}2_i$ . Compute the values

$$R1 = \frac{NS_{C_j} d(x_i, C_j)^2}{NS_{C_j} - 1}$$

and

$$R2 = \frac{NS_{C_l} d(x_i, C_l)^2}{NS_{C_l} + 1}.$$

If  $R1$  is less than  $R2$ , object  $x_i$  remains in cluster  $C_j$ . Otherwise, switch  $\bar{C}1_i$  and  $\bar{C}2_i$  and update the centres of clusters  $C_j$  and  $C_l$ .

**Step 7.** If no transfer took place in the last  $N_s$  steps, go to **Step 4**, otherwise go to **Step 6**.

### 7.6.4.3 Clustering Criteria

There are a large number of stopping rules that can be used in  $k$ -means clustering, to determine the optimal number of clusters. These can be divided into three categories (Dimitriadou et al., 2002).

- Those based on the within-clusters sum of squares (SSW) and the between-clusters sum of squares (SSB). Such criteria are, among others, the *Calinski-Harabasz* and *Hartigan* measures.

- Criteria based on the statistics of the scatter matrix of the data, and the sum of the scatter matrices in every cluster. Criteria in this category, include among others, the *Scott* and *Friedman* measures.
- Criteria that do not belong to the two previously mentioned categories. Such criteria are the *Simple Structure Index* (SSI) and the *Negative Log-Likelihood*.

In this project, four criteria, will be used, the *Calinski-Harabasz*, the *Ratkowsky-Lance*, *Hartigan's* criterion and *Trace W*, to ensure a representation of a range of criteria for better determination of the optimal number of clusters. The optimum number of clusters will be determined by examining the following aspects of the values of the criteria, in a plot of criterion value vs the number of clusters (see Figure 7.21):

- The maximum or minimum value of the criterion ( $\max_k i_k$  or  $\min_k i_k$ , respectively, with  $k$  being the number of clusters and  $i_k$  the criterion value for  $k$  clusters).
- The maximum difference from the cluster at the right side of the plot,  $\max_k (i_k - i_{k+1})$ , where the curve has its maximum decrease.
- The maximum difference from the cluster at the left side of the plot,  $\max_k (i_{k+1} - i_k)$ , where the curve has its maximum increase.
- The maximum or minimum value of the second differences,  $\max_k ((i_{k+1} - i_k) - (i_k - i_{k-1}))$ , where the curve has an elbow.

The four chosen criteria are described in detail below:

### Calinski-Harabasz

This index is computed as

$$\frac{SSB/(N_c - 1)}{SSW/(N_s - N_c)}$$

where  $N_s$  and  $N_c$  are the total number of objects and number of clusters, respectively (Milligan and Cooper, 1985). The optimal number of clusters is that for which this index obtains its maximum value.

### Ratkowsky-Lance

This is given by

$$\sqrt{\frac{\text{varSSB}}{\text{varSST}}} \frac{1}{N_c}$$

where  $\text{varSSB}$  and  $\text{varSST}$  are the between-clusters and total sum of squares respectively, for each dimension in the data, and  $N_c$  is the number of clusters (Milligan and Cooper, 1985). The optimal number of clusters is given by considering the two points that give the maximum value of the difference of a point from the point at its right side in the plot and choosing the number which gives the highest value of Ratkowsky-Lance's index.

## Hartigan

This index was proposed by [Hartigan \(1975\)](#) as

$$\log \left( \frac{SSB}{SSW} \right)$$

based on the sum of squares. The optimal number of clusters is given by considering the two points that give the maximum value of the difference of a point from the point at its left side in the plot and choosing the number which gives the lowest value of Hartigan's index.

## Trace $W$

This is one of the most popular criteria in clustering analyses ([Milligan and Cooper, 1985](#)).  $W$  is the within clusters covariance matrix. The optimal number of clusters is given by considering the three points that give the maximum value of the second order differences of sequential values of the criterion in the plot and choosing the number which gives the highest value of Trace  $W$ . The lower the value of this criterion, the more homogeneous (compact) the clusters are ([Dimitriadou et al., 2002](#)).

## 7.6.5 Application of the $k$ -means Algorithm to the Data

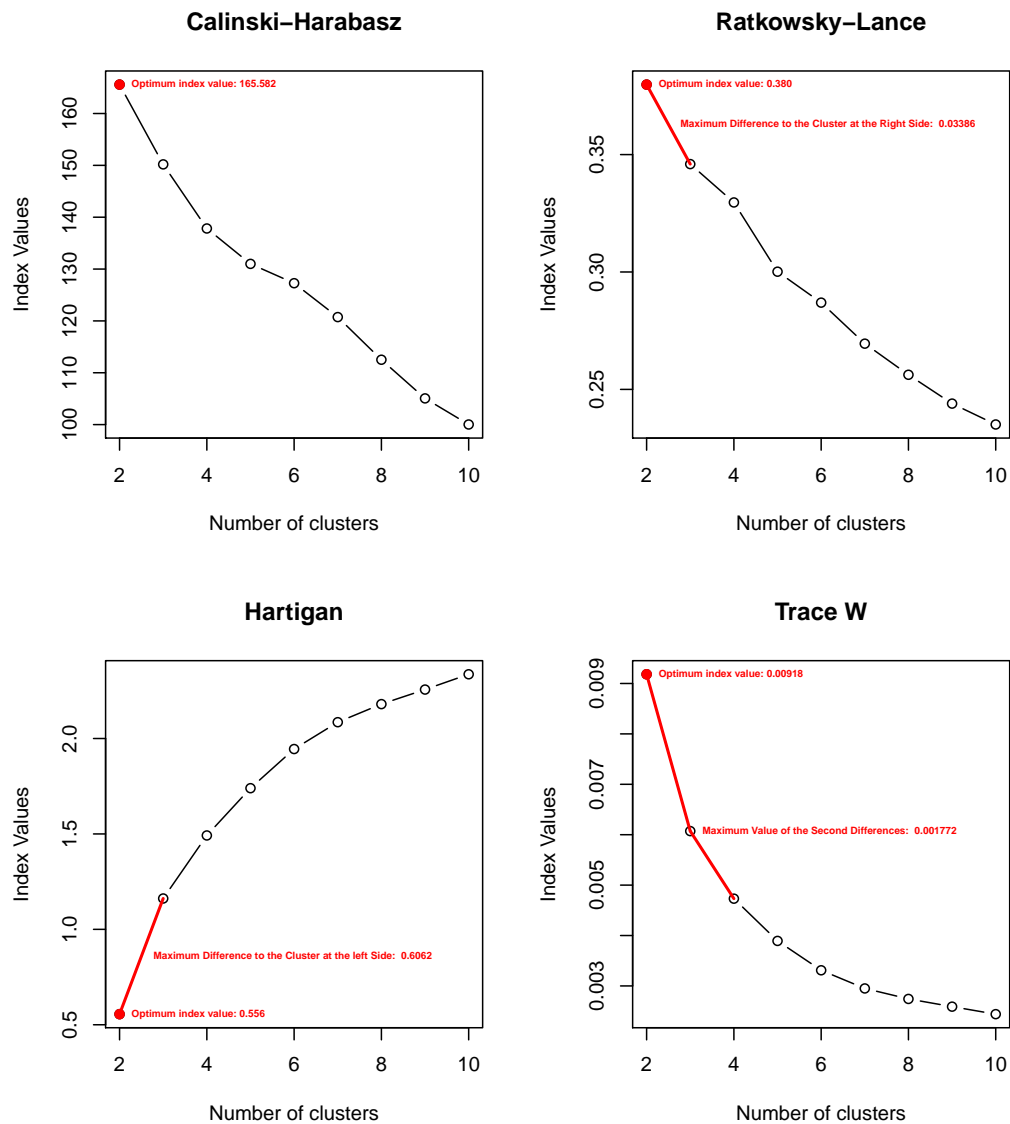
### 7.6.5.1 Introduction

The data that will be analysed by  $k$ -means clustering methods is the same that was used in the HCA and fuzzy clustering analyses. The data set includes the 97 patients with specific *response to AEDs* information (only responders or non-responders), with intensity values in the proton NMR chemical shift range of 5.98 – 0.02 *ppm*. The data has also been row-scaled to a constant total before analysis.

### 7.6.5.2 Determination of the Optimal Number of Clusters

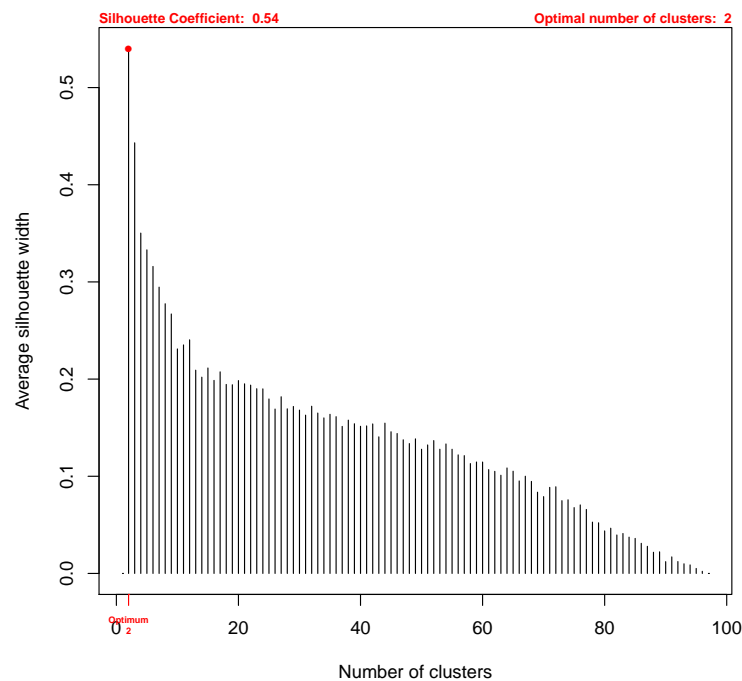
The algorithm described in Section 7.6.4.2 will be used to provide a  $k$ -means clustering method. To improve the chances of the algorithm converging to the global solution, in all analyses 1000 random sets of distinct rows of the data matrix were chosen as the initial centroids of the clusters. To determine the optimal number of clusters, four different criteria will be used in the derived  $k$ -means partitions of 2-10 clusters. These criteria are the *Ratkowsky-Lance*, the *Calinski-Harabasz*, *Hartigan* and *Trace  $W$* . Figure 7.21 shows the results for the four criteria. In red is shown the point which corresponds to the optimum number of clusters. All four criteria point to an optimum solution of 2 clusters.

To confirm the results of the criteria, the silhouette values for the  $k$ -means partitions of 2-96 clusters will be computed. Figure 7.22 gives the overall *average silhouette widths*



**Figure 7.21:** Values of four clustering criteria for  $k$ -means partitions of 2-10 clusters. The red point represents the optimum number of clusters. The red segments indicate the maximum difference between two clusters (Ratkowsky-Lance and Hartigan) and the maximum second differences (Trace W).

for all partitions of 2-96 clusters. In red is depicted the optimal number of clusters. As can be seen, the *silhouette coefficient* is 0.54, as this is the highest *average silhouette width* and is derived from the 2-cluster method. From the silhouette information and the criteria values, it can be concluded that the optimum number of clusters is 2. Although the 2-cluster  $k$ -means partition has proved to be the most appropriate to fit the epilepsy data, it remains to be seen whether this clustering method can also discriminate the patients with respect to their clinical characteristics.



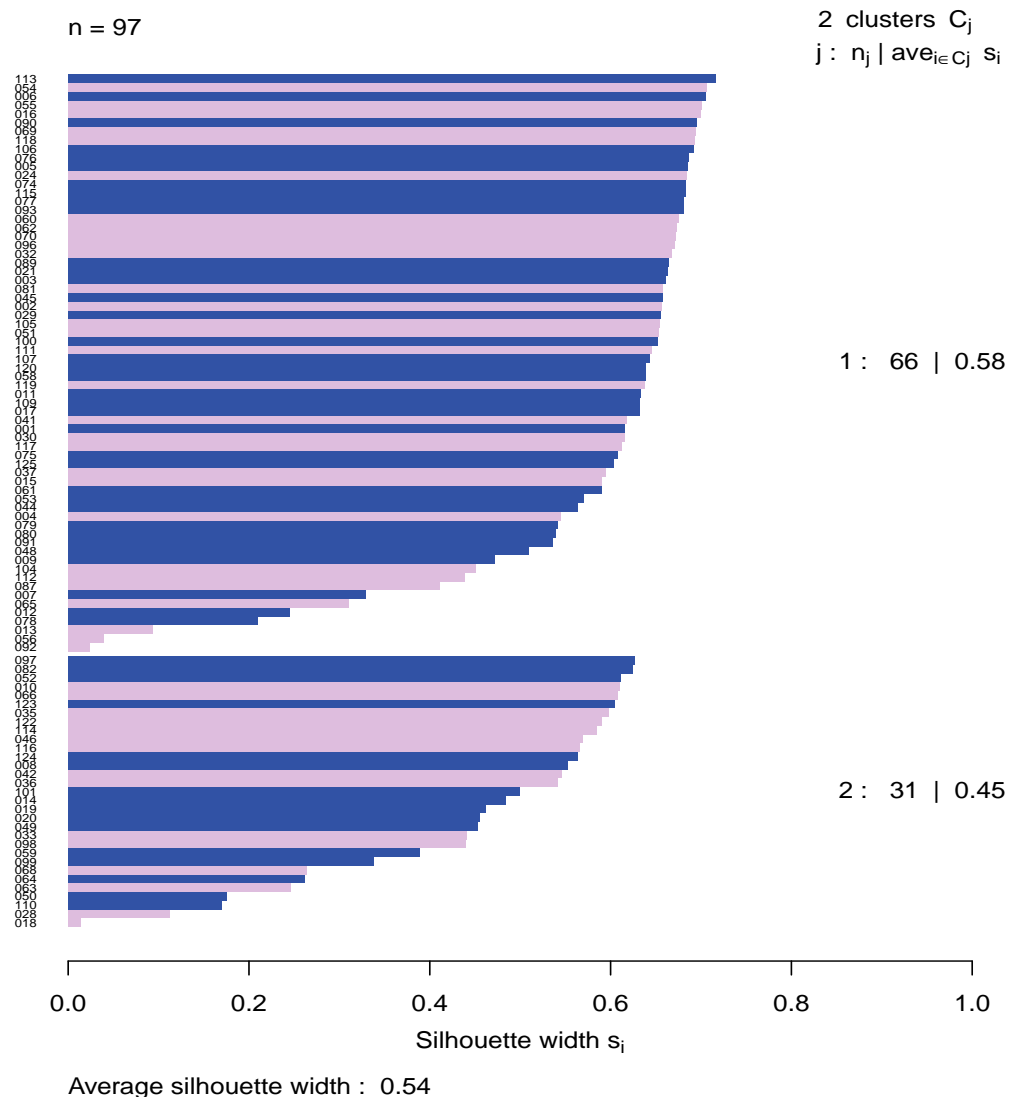
**Figure 7.22:** Average silhouette widths for partitions of 2-96 clusters for the selected  $k$ -means clustering method. The optimal number of clusters is indicated in red.

### 7.6.5.3 Discrimination of the Clinical Characteristics

A *silhouette plot* for the 2-cluster partition can be seen in Figure 7.23. The colour in the bars corresponds to the *response to AEDs* categories, with blue and pink representing responders and non-responders, respectively.

The *average silhouette width* values for clusters 1 and 2 are 0.58 and 0.45, respectively, and the *average silhouette width* for the entire data set is 0.54. Although these values are lower than those obtained from fuzzy and HCA clustering, in the  $k$ -means clustering solution there are no misclassified patients. Patients 13, 44 and 70 have very low silhouette values in cluster 1, whereas patients 18 and 23 have similarly low values in cluster 2. Patients 13 and 18 in particular have been identified in all three clustering methods as weakly classified or misclassified in the two clusters, therefore it looks as if these patients are either outliers or possibly belong to a third cluster.

The *silhouette plot* also shows that the first and second clusters contain 66 and 31 patients respectively, exactly as in the fuzzy clustering case. However, the  $k$ -means clustering method being discussed does not fit the epilepsy data as well, as the fuzzy clustering method (having an overall *average silhouette width* of 0.74). It is however, approximately as good as the HCA method is in fitting the data, as the HCA method's overall *average silhouette width* of 0.59 is quite close to that of the  $k$ -means clustering method (0.54). Comparing the  $k$ -means with the HCA clustering method, the patients



**Figure 7.23:** *Silhouette plot* for the 2-cluster partition derived by the  $k$ -means clustering method. The blue and the pink bars correspond to responders and non-responders respectively. The *average silhouette width* for clusters 1 and 2 is 0.58 and 0.45 respectively, and the *average silhouette width* for the entire data set is 0.54.

in the  $k$ -means partition are slightly better fitted as cluster 2's *silhouette widths* are 0.45 and 0.43 for  $k$ -means and HCA respectively. In addition, the plot confirms that there is no discrimination of the patients with low misclassification rate, with respect to their *Response to AEDs*.

The cross-tabulation of *Response to AEDs* information with the clusters of the selected  $k$ -means clustering method can be seen in Table 7.21. There are  $16 + 30 = 46$  misclassified patients in the  $k$ -means method, which is slightly worse than for HCA and as good as the fuzzy clustering method, having 45 and 46 misclassified patients, respectively. Considering the results in Tables 7.18 and 7.21, it seems that the optimal

**Table 7.21:** Cross-tabulation of the optimal 2-cluster  $k$ -means partition to *Response to AEDs*.

Clusters	Response to AEDs	
	Responder	Non-responder
1	36	30
2	16	15

partitioning fuzzy and  $k$ -means clustering methods are quite similar in capability with respect to the allocation of patients to the two clusters in their partitions. However, the *Response* results in both methods show that they cannot discriminate the patients with respect to their *Response to AEDs* with low misclassification rates. The derived optimal 2-cluster  $k$ -means partition is the same as was obtained by the optimal fuzzy clustering partition, therefore the results of any extra analysis is the same as in the fuzzy clustering case.

The clinical characteristics' cross-tabulation with the two clusters of the optimal  $k$ -means method, being the same as for fuzzy clustering, can be seen in Tables 7.19 for *Gender* and *Age*, and 7.20 for *Seizure type* and *BMI*. Results of the two statistical tests of the relationship of the clinical characteristics to the two clusters of the partition, similarly to fuzzy clustering, shows that there is no relationship for *Response to AEDs*, *Gender* and *Seizure type*, whereas there is a relationship for *Age* and *BMI*.

Figure 7.18, for fuzzy clustering, illustrates also the clustering solution derived by the 2-cluster  $k$ -means method. That is, a two-dimensional projection of the clustering solution is drawn, such that the first two principal component scores (according to the results from Chapter 5) can be seen superimposed with the partition derived by the selected  $k$ -means clustering method. In both scores plots, black and red represent the patients clustered to the first and second cluster respectively. The bottom scores plot illustrates the *Response to AEDs* information, such that points labelled as "R" and "N" correspond to responders or non-responders to AEDs respectively. Similarly, Figures 7.19 and 7.20 illustrate the component scores superimposed with the *Age* and *BMI* information, respectively. Similarly to fuzzy clustering, both of the clusters are not compact. Also, there is no distinction among responders and non-responders to AEDs by this clustering. Therefore, as in the fuzzy clustering and HCA methods, this algorithm has not been efficient in classifying patients according to their *Response to AEDs*.

The 2-cluster  $k$ -means partition is exactly the same as the 2-cluster fuzzy partition with regards to the patients contained in the two clusters of the partitions. The only thing which differs is the *silhouette width* of the patients in the two methods.



## 7.7 Competitive Learning Algorithms

### 7.7.1 Introduction

The general idea behind these algorithms is simple. Given a set of representatives  $w_i$ ,  $i = 1, \dots, m$ , when an input vector  $\mathbf{x}$  is presented to the algorithm, all  $m$  representatives compete with each other and the winner is the representative which is closer (with respect to some distance measure) to  $\mathbf{x}$ . The winner is updated to move it closer to  $\mathbf{x}$ , while the representatives that lost either remain unchanged or are updated to be closer to  $\mathbf{x}$  but at a much slower rate (Theodoridis and Koutroumbas, 2003). An important competitive learning algorithm is Kohonen's self organizing maps which will be described in the following section.

### 7.7.2 Self Organizing Maps

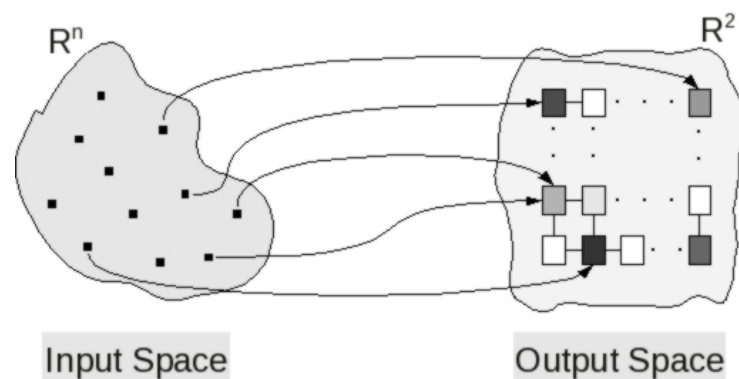
#### 7.7.2.1 Introduction

Kohonen's Self Organizing Map (SOM) is a statistical approach for cluster analysis and data visualization. It is based on the Artificial Neural Networks (ANN) learning technique (Izenman, 2008) as this was used to create simplified models of the way the human brain and its neural paths work (Taner, 1997). SOMs were introduced by Kohonen in the 1970s, and the initial algorithms have since been improved in many ways such as those described in Dittenbach et al. (2002); Wang et al. (2002); Jin et al. (2004); Wang et al. (2005); Salas et al. (2007), to accommodate various types of input data and areas of application. Areas of application include, among others, the selection of representative species in multivariate ecological data (Park et al., 2006), the analysis of metabolic profiles of patients with various diseases such as coronary heart disease (Suna et al., 2007) and type I diabetes (Makinen et al., 2008), and even the analysis and solution of water resources problems (Kaltch et al., 2008).

The SOM approach uses unsupervised learning to produce a mapping of a high-dimensional input space onto a two or three-dimensional output space, while it preserves the topological relationships between the input data elements as closely as possible (Dittenbach et al., 2002; Brereton, 2009). The SOM algorithm attempts to find clusters such that any two clusters that are close to each other in the output space have representatives close to each other in the input space. In the area of chemometrics and metabonomics, it can be used to provide information about the relationship between samples, to visualize characteristic variables, specific samples or groups of samples (Suna et al., 2007; Makinen et al., 2008).

The visualization of an input space in SOM consists of a grid of a large number of interconnected nodes. A SOM plot in two dimensions contains nodes usually arranged

as a square, rectangular or hexagonal grid (Figure 7.24). The size of an SOM grid is



**Figure 7.24:** Visualization of SOM data projection - 2D rectangular grid.

selected before the analysis by the researcher. That is, initially the number of required nodes in the grid is not usually known, therefore the total number of nodes is usually selected to be much larger than the suspected number of clusters in the data. After trial and error, the SOM grid can be adjusted, reducing the total number of nodes in the grid to a more representative size. Each node in the grid is associated with a representative (or prototype or codebook vector) in the input space, say  $w_c$ . This is a weight vector, such that the number of weights for each node (map unit) corresponds to the number of variables measured in the original data space. Therefore, this layer of weights for each variable can be considered as the third dimension of the SOM grid. Initially the components of all  $w_c$  vectors are set to be random numbers, using a random number generator which takes values within the range of the observed data. There are two main versions of the classic SOM algorithm, the on-line and the batch SOM (Izenman, 2008; Silva and Marques, 2007). In the former case, the input vectors are presented to the algorithm sequentially (one at a time and usually in random order), whereas in the latter case all input vectors are presented together at a time.

### 7.7.2.2 Classic On-line SOM Algorithm

The algorithm to follow in order to run an on-line SOM analysis can be summarized by the following steps:

#### Identification of Best Matching Unit

- An initial representative is chosen. Some of the choices that can be made include random samples from the data, random vectors from a  $N(0,1)$  distribution and the linear grids from the directions of the first two principal components of the PCA of the data. Usually a sample vector,  $x_k$ , is selected randomly (with or without replacement) from the data.

- Next step is the computation of the dissimilarity distance  $d_{(x_k, w_{ij})}$  between the selected random sample  $x_k$  and each of the map units' weights  $w_{ij}$ . The most common distance metric used for that purpose is the Euclidean distance measure

$$d_{(x_k, w_{ij})} = \|x_k - w_{ij}\|^2 = \sqrt{\sum_{j=1}^n (x_{kj} - w_{ij})^2}$$

where  $x_{kj}$  is the value of variable  $j$  for sample  $x_k$ ,  $n$  the total number of samples and  $w_{ij}$  is the weight of the  $j$ th variable for map unit  $i$ . The map unit with the smallest  $d_{(x_k, w_{ij})}$  for vector  $x_k$  is then located and called as the Best Matching Unit (BMU),  $U_{BMU}$  of sample  $x_k$  for the current iteration. That is,

$$d_{(x_k, w_{BMU})} = \min_k \{d_{(x_k, w_{ij})}\}$$

where

$$BMU = \arg \min_k \{d_{(x_k, w_{ij})}\}.$$

### Identification and Weights' Updating of Map Units

- To identify which map units are closer to the BMU, it is necessary to introduce the concept of neighbourhood and neighbouring units. A map unit  $\hat{U} \in M$  ( $M$  is the set of units in the map) is defined as a neighbour of the BMU unit  $U_{BMU}$  if the Euclidean distance of their codebook vectors  $w_{\hat{U}}$  and  $w_{BMU}$  is smaller than a predefined threshold  $\eta$  called the neighbourhood width. That is, the neighbourhood set  $N_{BMU}$  of BMU contains those units which satisfy the following:

$$N_{BMU} = \{u : d_{(w_{BMU}, w_u)} < \eta\}.$$

- Only the weight vectors of those units which belong to the neighbourhood set of BMU will be updated, using a distance-weighted formula such as the following

$$w_u = w_u + \alpha \eta_f(x_k - w_u)$$

where  $\alpha$  is a learning rate function and  $\eta_f$  is a neighbourhood function. The learning rate function indicates how much the selected map unit will be updated in each iteration, to approximate a sample as much as possible. There are many such functions (Izenman, 2008) and the most popular can be seen below:

$$\text{Exponential : } \alpha_i = \alpha_0 e^{-\frac{i \ln(\eta_0)}{I}}$$

$$\text{Linear : } \alpha_i = \alpha_0 \left(1 - \frac{i}{I}\right)$$

$$\text{Power : } \alpha_i = \alpha_0 \left( \frac{0.005}{\alpha_0} \right)^{\frac{i}{I}}$$

$$\text{Inverse : } \alpha_i = \frac{\alpha_0}{\left( 1 + \frac{100i}{I} \right)}$$

where  $i$  is the current iteration of the algorithm,  $\alpha_0$  the initial learning rate,  $\eta_0$  the initial neighbourhood width and  $I$  the total number of iterations. All the above mentioned functions ensure that the learning rate will monotonically decrease until the end of the training. A neighbourhood function is used to update the neighbourhood width in every iteration. Usually  $\eta$  has a large value initially, but as training continues it decreases monotonically until at the end of the training only the BMU and its adjacent units (adjacent neighbours) are updated. A number of neighbourhood functions that can be used in a SOM analysis are given below:

$$\text{Exponential : } \eta_i = \eta_0 e^{-\frac{i \ln(\eta_0)}{I}}$$

$$\text{Gaussian : } \eta_i = e^{-\frac{|w_u - w_{BMU}|^2}{2r_\eta^2}}$$

$$\text{Square (or bubble) : } \eta_i = \begin{cases} 1, & \text{if } |w_u - w_{BMU}| \leq r_\eta \\ 0, & \text{if } |w_u - w_{BMU}| > r_\eta \end{cases}$$

where  $r_\eta$  is the neighbourhood radius. The algorithm is executed until the total number of iterations  $I$  is reached. A recommended value for  $I$  is 500 times the number of units in the map, for  $\eta_0$  half width of the map and for  $\alpha_0$  0.1 (Brereton, 2009). It is also useful sometimes to perform the training using two stages of the algorithm. At stage 1, the initial learning rate is large, e.g. 0.1, whereas at stage 2 fine-tuning is performed using a much smaller value, e.g. 0.01, for the initial learning rate, as the map has already been trained at the first stage. Sometimes the  $k$ -means algorithm is used as the fine tuning stage (Brereton, 2009).

### 7.7.2.3 Classic Batch SOM Algorithm

In batch SOM, updates of the weight vectors occur only at the end of each learning epoch (the presentation of the whole training data to the algorithm). The new weights,  $w_{u'}$  can be computed using the following equation:

$$w_{u'} = \frac{\sum (\eta_{f'} x_k)}{\sum \eta_{f'}}$$

and winner unit (BMU) can then be found using equations

$$d_{(x_k, w_{u'})} = \|x_k - w_{u'}\|^2$$

and

$$d_{(x_k, w_{BMU})} = \min_k \{d_{(x_k, w_{u'})}\}.$$

This procedure is repeated from the beginning until the selected convergence criterion is met. There is no change in the neighbourhood function from that in the on-line algorithm, but the learning rate function is not used in batch SOM, thus reducing significantly the risk of poor convergence. Due to the way that the batch algorithm is executed, it is much faster than the on-line version. It is also clear from the algorithm that the batch SOM requires the whole set of input vectors during the training procedure. In addition, as the weight vectors are updated after an epoch, the order in which the input vectors are presented is of no importance and the last input vectors do not influence the final results.

#### 7.7.2.4 Goodness of Mapping

Self-organizing maps aim to preserve the topological information of the input space (Villmann et al., 1997). Due to the fact that SOM is a vector quantization algorithm (Vesanto, 1999), the projection of a multi-dimensional input space to two or three dimensional output spaces, as is usually the case in biological data spaces, can affect the goodness of the mapping. Although the probability distribution of the input space is usually depicted adequately in SOM algorithms (Kiviluoto, 1996), there are other aspects of the mapping that need to be considered in order to establish that the mapping is of good quality and that it preserves the input space topology (Brereton, 2009). Criteria to evaluate the goodness of a SOM mapping include the mapping continuity (Neme and Miramontes, 2005; Kiviluoto, 1996) and resolution (Polani, 1999; Brereton, 2009). As Kiviluoto (1996) states, when a mapping is continuous any samples that are close in the input space are mapped close to each other in the output space as well, while a mapping of good resolution means that no samples that are distant in input space are mapped close in the output space. There are many goodness of fit measures in SOM (Pözlbauer, 2004; Bauer et al., 1999). The following are the most commonly used in chemometrics and metabonomics studies:

##### Mean Quantization Error (MQE)

This is a quality measure which can be applied to any form of vector quantization and clustering algorithm (Pözlbauer, 2004). In general, it represents the average distance of the sample vectors to the cluster centroids they belong to. In SOM, it is the average distance of each sample to the representative of its Best Matching Unit after the last of

the iterations of the SOM training algorithm has been completed. This can be calculated as

$$MQE = \frac{1}{n} \sum_{k=1}^n d_{(x_k, w_{BMU})}.$$

MQE cannot be used to compare SOMs of different grid sizes, since the measure decreases monotonically as the map size increases. In addition, MQE depends on the initialization procedure and on the training data, and gives the best results when the number of map units is at least as large as the number of training samples (Brereton, 2009).

### Topographic Error (TE)

Topological preservation can be measured by using this measure. More specifically, it measures the continuity of the SOM mapping. If for a sample  $x$  the closest and second closest representatives represent adjacent map units, the map is locally continuous, otherwise there is a local topographic error (Pözlbauer, 2004). Summing up and normalizing the number of local topographic errors for all samples gives the topographic error for the whole mapping (Kiviluoto, 1996). The topographic error is given (Neme and Miramontes, 2005; Villmann et al., 1997) by the formula:

$$TE = \frac{1}{n} \sum_{k=1}^n \theta_{(x_k, w_u)},$$

where  $u = 1, \dots, U$  and

$$\theta_{(x_k, w_u)} = \begin{cases} 1, & \text{if } \forall i \exists j, k : i \in \\ & \{1, \dots, j-1, j+1, \dots, k-1, k+1, \dots, u\} \\ & \|u_j - x_i\| \leq \|u_k - x_i\| < \|u_l - x_i\|, |j-k| > 1 \\ 0, & \text{otherwise} \end{cases}$$

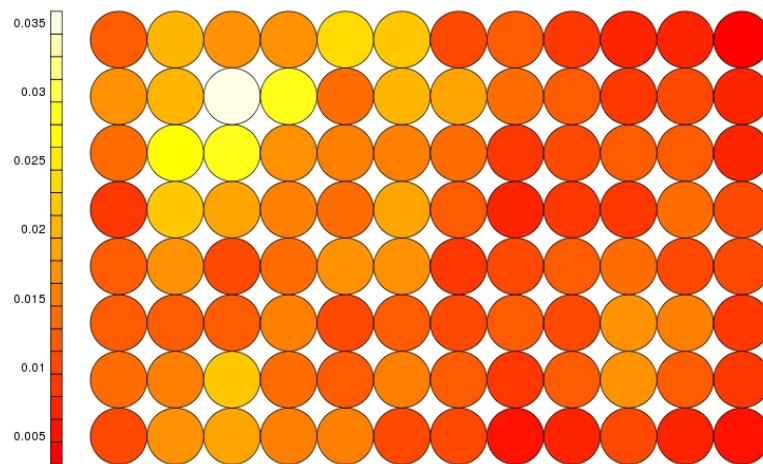
with  $u_i$  being the representative vector of unit  $i$ , and  $U$  the last unit in the map. Although the topographic error indicates the proportion of local neighbourhoods that are mapped correctly, it cannot describe the type of local discontinuities that may appear in the map (Kiviluoto, 1996). A large number of discontinuities in the map means that the topology of the input space has not been preserved and the mapping is not accurate. Similarly to MQE, the TE depends on the input data and the training parameters, giving better results when the map is overfitted.

#### 7.7.2.5 Means of Visualization

There are many ways to illustrate the results of a SOM analysis. The most useful are the following:

### Unified Distance Matrix (U-matrix)

A U-matrix is a visualization tool which allows the identification of any clustering in the map by using a representation such as colour-coding of the Euclidean distance between the neighbouring representatives in the input space. Small values indicate that the nodes in the map are quite close in similarity not only in output space but also in input space, whereas larger values indicate that the representatives are not neighbours in the input space (Brereton, 2009). An example of a U-matrix for the epilepsy data using a grid of size  $12 \times 8$  can be seen in Figure 7.25. In this example, for instance, the three

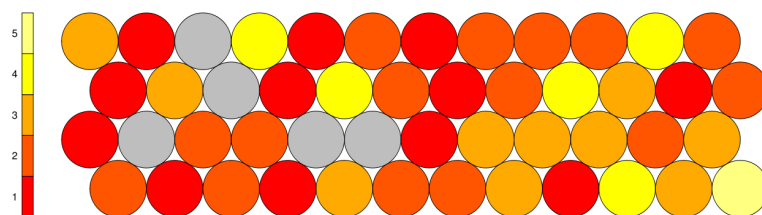


**Figure 7.25:** Example of a Unified Distance matrix.

yellow units contain samples which are far away from each other in the input space, as their values are very large ( $> 0.025$ ), in comparison to the dark red units at the top-right and bottom -right of the map, which have very small values ( $< 0.005$ ).

### Hit Histogram

This type of plot is used in order to visualise the Best-matching Unit for each sample and at the end of the training. Each unit has a value showing the number of times that the map unit was the Best-matching Unit of any sample at the end of the training (Brereton, 2009). The visualization can be done in two or three-dimensional histograms. A two-dimensional histogram illustrates the number of BMU hits by the size of the shaded map units (or by colour-coding the hits), whereas in a three-dimensional histogram the height of each hexagon bar is proportional to the number of hits. Ideally, in a case of several classes present in a dataset, only one or a small number of map units should correspond to the BMU of all samples from the same class and therefore correspond to a high number of hits. Map units with high numbers of hits usually suggest a concentration of samples around these units and these units are mainly on the periphery of a SOM map. In addition, map regions with a large number of hits correspond to map regions of similarity shown in the U-matrix plot. An example of a 2D colour-coded hit histogram for a grid of size  $12 \times 4$  for the epilepsy data, can be seen in Figure 7.26. For instance,

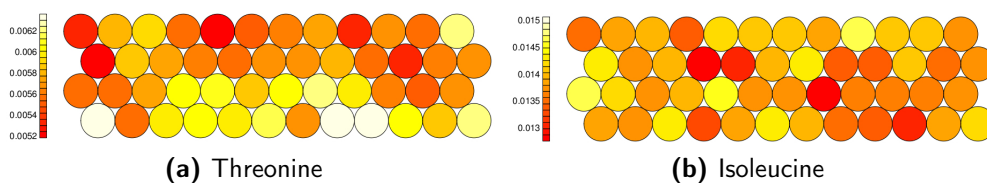


**Figure 7.26:** Example of a hit histogram.

the bottom-right unit in the hit histogram has a value of 5, meaning that this specific unit was the best-matching unit of any sample five times when the training finished. The grey units do not contain any samples. The six yellow units (having four or five hits each) in the example map, could suggest the existence of six clusters lying around these units.

### Component Planes

The above mentioned visualization tools cannot illustrate the importance of the variables in the input space. In a similar way to PCA, it might be useful to explore the contribution of variables to the map. More specifically, component planes can illustrate how each input variable influences the map and the relationship among the variables and the samples in the data (Brereton, 2009; Makinen et al., 2008). A single component plane is constructed for each variable by converting the weights for a specific variable into colour-coding according to the importance for describing a given region to the map (such as in case of the U-matrix). In this way, the relationship between samples and variables is visualized by colouring each map unit,  $k$ , proportionally to the weight  $w_{kj}$  of unit  $k$  for the chosen variable  $j$ . The main interest in using component planes is to identify whether a variable can describe a class and not if it can discriminate between two classes. In the case of only two classes existing in the data, those variables that best describe a class are also good discriminators, but if more than two classes are indicated in the data, this fact is not necessarily true any more. For example, in Figure 7.27 the component planes of two human blood serum metabolites, *isoleucine* at 1.98 ppm and *threonine* at 4.26 ppm, can be seen in an SOM map of dimensions  $12 \times 4$  for the epilepsy patients. The planes show the magnitude of each map unit's weight for *isoleucine* and



**Figure 7.27:** Examples of component planes.

*threonine*. In map regions with darker shading, the variables have larger values and these regions correspond to regions with high similarity (in the context of the U-matrix



visualization). That is, a variable is associated more with those patients for whose, the map regions belonging to, are darker. So, in the given examples, in both metabolites, for instance, the dark-red coloured units, which have very small values ( $< 0.053$  for *threonine* and  $< 0.013$  for *isoleucine*), contain samples which are very closely associated with these two metabolites, whereas in the case of the white-coloured units in *threonine*, with values  $> 0.0062$ , the patients contained in these units are related in any way to this specific metabolite.

### 7.7.2.6 Application to the Epilepsy Data

#### Introduction

As the SOM maps that are produced are completely dependent on the input data used for the analyses and the various learning parameters, there are no standard rules established which produce a good quality map, i.e. a highly accurate and well-ordered map in every case. For example, in the case of epilepsy data, it is logical that it is probably more important to obtain a highly accurate map (as good representation as possible of the input space to the output space) than to preserve the topological order of the input space), whereas in a data-mining application the order would logically be more important than the accuracy as it commonly involves documents. Therefore, in the epilepsy case, the mean quantization error is probably more important than the topographic error. With these considerations in mind, the following SOM analyses and the selected parameters of the methods, were chosen to investigate whether the SOM algorithms can be used to identify any common patterns among patients with response or no response to AEDs, and not to find the best possible mapping of the data.

#### Initialization

Before the analyses, the samples were normalised to eliminate the possibility of any influence on the SOM results by any of the metabolites due to a metabolite's large variance or absolute value. The shape of the SOM grid was chosen to be hexagonal to avoid a preference of the SOM algorithm towards horizontal or vertical directions (Park et al., 2006). In addition, the size of the map must not be such that it has more units than samples in the data set, to ensure better response from the map quality criteria. The classic online SOM algorithm was used. The map size was determined using Vesanto's (Vesanto et al., 2000) heuristic formula which states that the total number of map units,  $N_U$ , is given by

$$N_U = 5\sqrt{n_s}$$

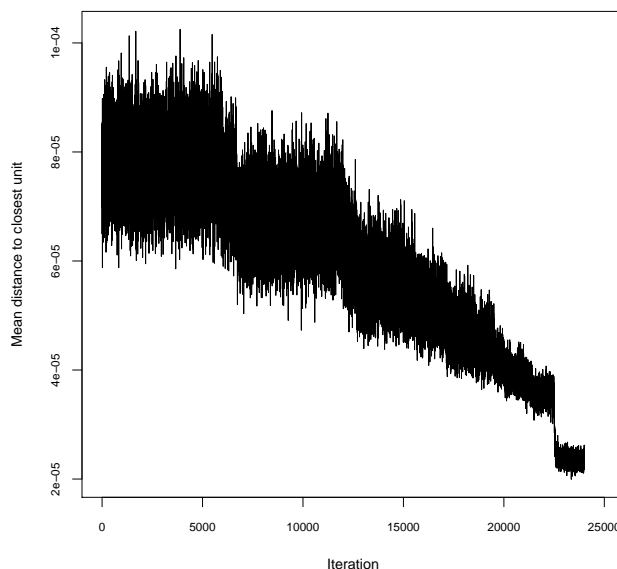
where  $n_s$  is the number of samples in the input space. The lengths of the grid sides can be calculated by setting the ratio of the lengths of the sides similar to that of the two

largest eigenvalues of the training data such that the product of the lengths is as close as possible to  $N_U$ . In our case,  $N_U$  is approximately 49 ( $5\sqrt{97}$ ), whereas the ratio of the two largest eigenvalues of the covariance matrix is approximately  $\frac{13}{1}$ , thus the grid dimension can be  $24 \times 2$  to approximate as closely as possible the map size without violating the ratio rule (Vesanto et al., 2000; Park et al., 2006).

The training parameters are the following: The learning rate initially has a value of 0.05 and decreases monotonically until it reaches the value 0.01 at the end of the number of epochs. The initial radius of the neighbourhood function is approximately  $\frac{2}{3}$  of the estimated map width, to allow for a large part of the map to be updated initially. A sufficient initial radius value is usually to cover  $\frac{2}{3}$  of all unit-to-unit distances. Thus, the initial radius for the  $24 \times 2$  grid is 16, whereas for the  $3 \times 2$  grid it is 2. The final value of the radius is 0 at the end of the algorithm. The initial representatives were chosen randomly without replacement from the data set.

### Training

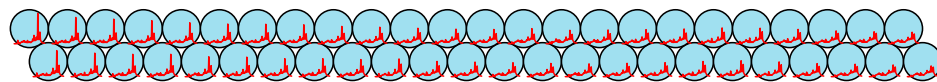
A series of runs was performed using the recommended values for the training parameters (Brereton, 2009; Tan and George, 2004). Thus, the total number of iterations was chosen in each case to be 500 times the map size. Two map sizes were used. Specifically the  $24 \times 2$  and the  $3 \times 2$  maps were chosen for further investigation. The neighbourhood width function converges to 0 after 24000 and 3000 iterations respectively for the estimated maps (Figure 7.28 shows the convergence for the  $24 \times 2$  grid). The quality



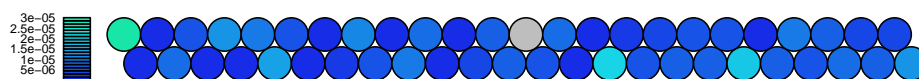
**Figure 7.28:** Convergence of the neighbourhood width function for the selected map ( $24 \times 2$  grid).

of the mapping can be examined by using specific plots to illustrate how closely to the codebook vectors in each unit the samples in the unit have been mapped. The mean

distance of samples mapped in each unit to the codebook vector of that unit can be illustrated using colour-coding such that the smaller the distances (darker colouring), the better the samples in the unit are represented by this unit's codebook vector. In Figure 7.29 the codebook vectors are illustrated beside the quality map for the samples. It can be seen that in general, the quality of the mapping is quite good, as in most of



(a) Codebook vectors



(b) Quality map

**Figure 7.29:** Illustration of the quality of mapping with regards to the samples. The grey unit in the quality map means that there is no sample mapped to this unit.

the map units the samples are quite close to the respective codebook vectors. However, clearly, three of the units have not been mapped accurately, with the worst approximation being in the top left-most unit. The two bottom right map units (in light blue colour) have also been mapped badly.

In addition, the Unified Distance matrix can be used to illustrate the average distance of each map unit to all immediate neighbour units. As is logical, units near a cluster boundary are expected to have higher average distances to their neighbour units (Figure 7.30). The black lines indicate a six-cluster solution using hierarchical clustering to allow



**Figure 7.30:** Unified Distance matrix for the  $24 \times 2$  grid.

comparison of the SOM clustering to the HCA solution. The units on the right side of the U-matrix are closer to each other than those on the left side of the matrix, however there is no indication from this matrix that there is a small number of clusters in the data according to the  $24 \times 2$  SOM analysis.

A two-dimensional colour-coded hit histogram for the SOM solution can be seen in Figure 7.31. The units with three or four hits indicate the existence of clusters in the areas surrounding these units, and the concentration of samples around these units is expected to be far larger than elsewhere in the map. Four such units are (2,1), (2,10), (1,13) and (1,23) where the first number corresponds to the row and the second to the

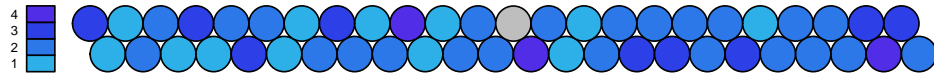
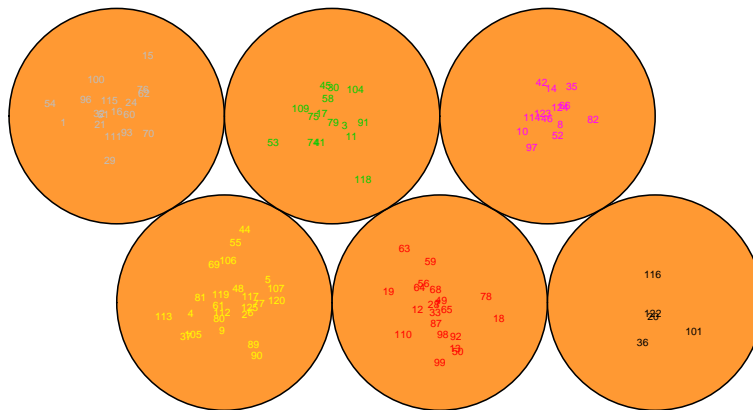


Figure 7.31: Hit histogram for the  $24 \times 2$  SOM solution.

column in which the unit is located in the map (with (1,1) being the bottom left-most unit and (2,24) the top right-most unit in the map). The results of the hit histogram for these four units correspond to those regions in the U-matrix with high similarity e.g. the single unit (2,1) at the top-left of the map and the 7 units at the right part of the U-matrix plot (as they are separated by the black lines of the six-cluster HCA partition), indicate the existence of clusters in these areas.

The mapping of the samples assigned to each of the six groups provided by the  $3 \times 2$  map, as well as the colour-coded samples map (using the corresponding label colours of the groups obtained by the  $3 \times 2$  map) for the  $24 \times 2$  grid can be seen in Figure 7.32. The cluster sizes of the above maps are 25, 21, 5, 18, 15 and 13 for groups 1-6 respectively.



(a) Samples -  $3 \times 2$  map



(b) Samples -  $24 \times 2$  map

Figure 7.32: Illustration of clustering the epilepsy data to six groups using SOM.

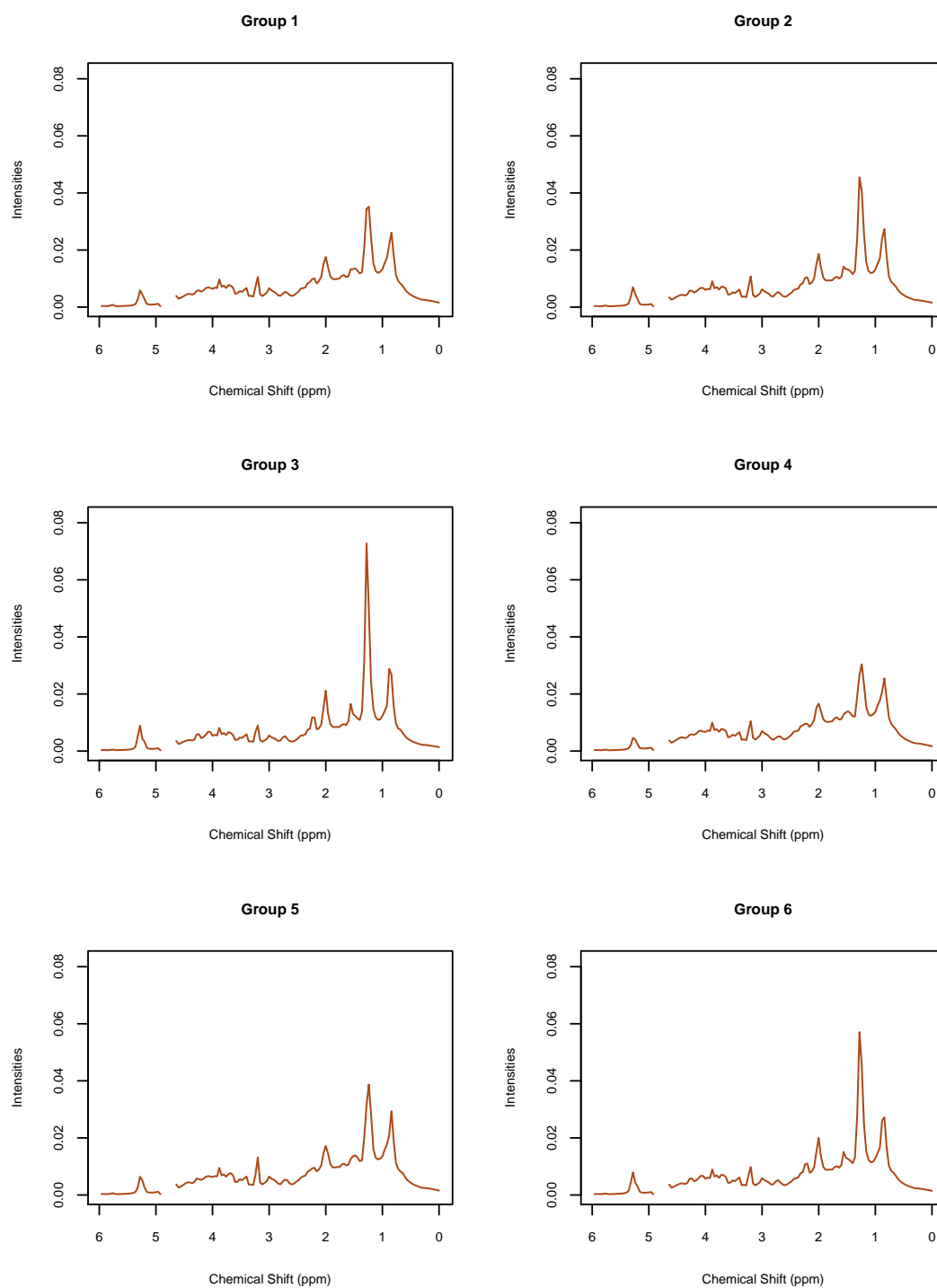
respectively. In the  $3 \times 2$  grid, group 1 corresponds to the bottom-left unit, while group

6 corresponds to the top-right unit counting from left to right and then from the bottom to the top row of the map. In the  $24 \times 2$  grid, the corresponding groups to the  $3 \times 2$  grid are 3, 6, 2, 1, 5, 4 from left to the right of the map. In the  $24 \times 2$  map, a number of units contain samples from more than one group of patients as were identified in the  $3 \times 2$  grid. Comparing the clustering results of the  $24 \times 2$  map to those of the hierarchical clustering in the U-matrix plot (Figure 7.30), it is clear that only cluster 4, at the right-most side of the map, is identical in both solutions, with the other clusters having slight differences (e.g. cluster 2, the third cluster from the left in the map) or large differences (e.g. clusters 1 and 5 at the middle - right part of the map have been merged to one cluster in the hierarchical clustering solution).

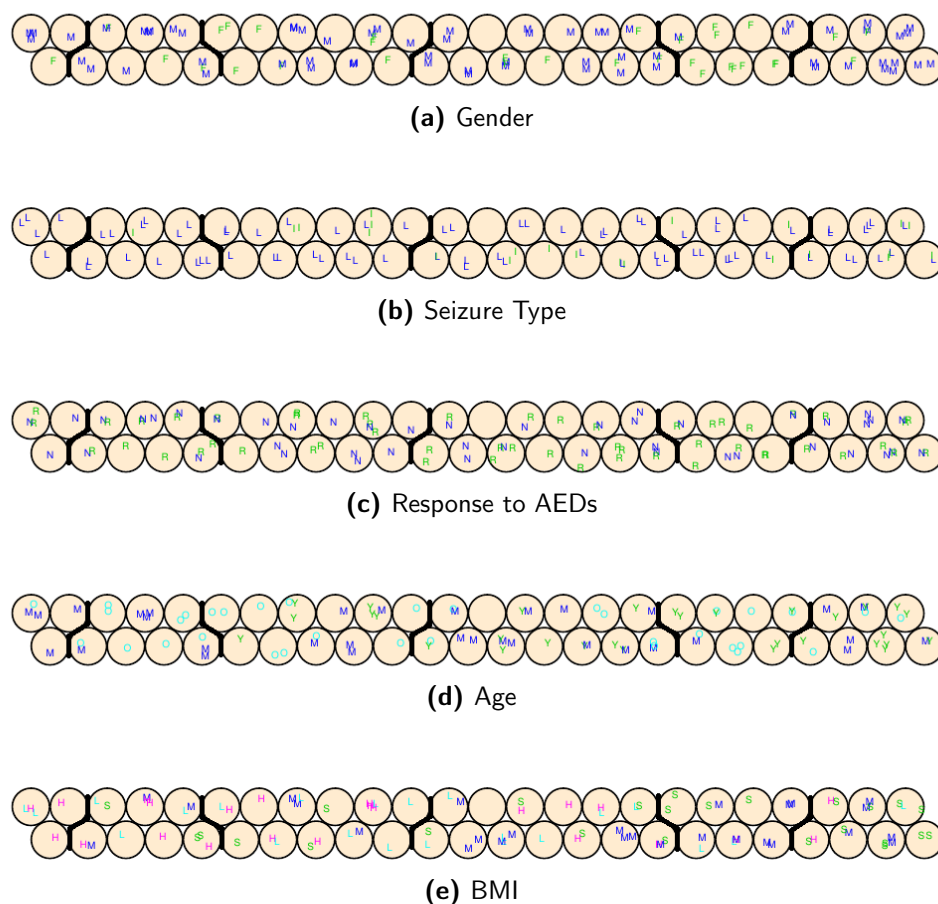
The mean spectra of the six groups can be seen in Figure 7.33. The main differences in the intensity levels of the variables in the six groups can be isolated in the *ppm* areas of  $\approx 5.5 - 5.1$ ,  $\approx 3.5 - 3.1$ ,  $2.1 - 2$  and  $1.6 - 1$  with the more important area being the last one, as the variables lying in this area have the largest intensity values. This is most significant around  $1.3 \text{ ppm}$ , where the mean intensity values for groups 3 and 6 are approximately 0.08 and 0.06 respectively, whereas group 4's mean value is approximately 0.03 (being the smallest of all six groups of patients).

To investigate whether there are any common patterns in the data with regards to the clinical characteristics of the patients, mappings of the samples with the clinical characteristics colour-coded have been plotted (Figure 7.34). The labels used in the *Gender* plot are *F* and *M* for *Female* and *Male* respectively. In the case of *Seizure type*, *I* means *IGE*, whereas *L* means the *LRE* type of seizure. Concerning the *Response to AEDs*, *R* stands for *Responder* (improvement to the patient's seizures) and *N* stands for *Non-responder* (no improvement to the patient's seizures). *Age* has three labels, namely *Y*, *M* and *O* for the three *Age* categories (*Young*, *Middle* and *Old* respectively). The *BMI* categories mentioned in Chapter 2, Section 2.4.3, represented by *Small*, *Medium*, *Large* and *Huge*, have the labels *S*, *M*, *L* and *H* in the *BMI* plot. An initial assessment of the clinical characteristics in Figure 7.34 shows that groups 3, 4 and 6 are dominated by males, and group 5 by females. In all groups, *LRE* is the dominating type due to the fact that there are many more *LRE* than *IGE* patients, whereas with respect to *Response to AEDs*, the groups are more balanced with only group 5 being dominated by responders to AEDs. Groups 3 and 6 are dominated by middle-aged and old age patients, whereas in group 4 the majority of patients are young. Large or huge *BMI* is observed in group 3, but not in groups 4 and 5 where the majority of patients have small or medium *BMI*.

To examine how the variables influence the map and what is the relationship between each variable and the samples in the data, component planes have been created for a selected number of variables. The maps for the ten variables with the largest mean values can be seen in Figure 7.35. Most of these variables are also in the ten variables



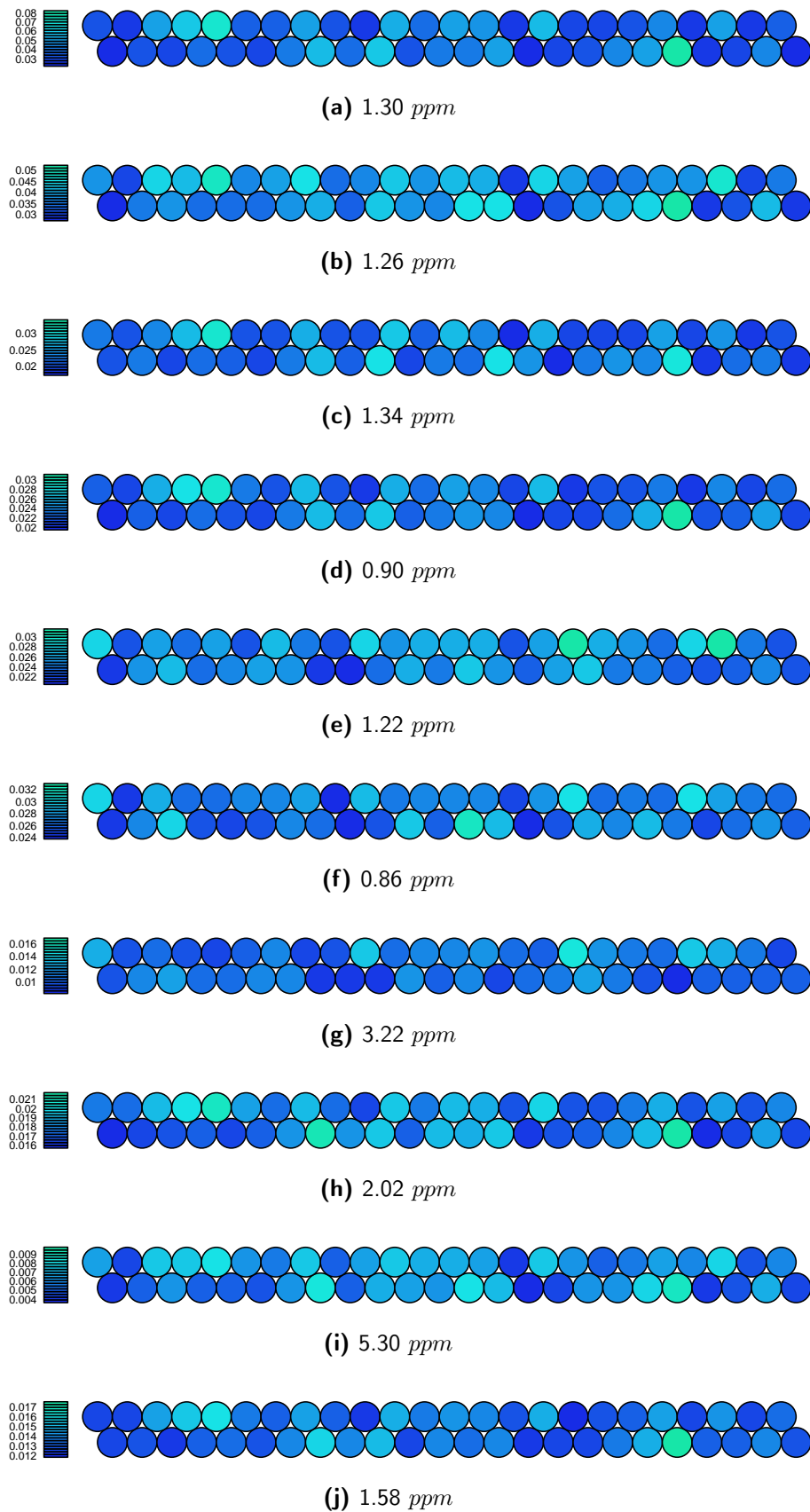
**Figure 7.33:** Mean spectra for the six groups of patients.



**Figure 7.34:** Illustration of the mapping of the samples according to the patients' clinical characteristics. The bold lines divide the map units to the six clusters in the SOM partition of Figure 7.32.

with the largest variance. These variables are at the spectral areas with chemical shifts 1.3, 1.26, 1.34, 0.9, 1.22, 0.86, 3.22, 2.02, 5.3 and 1.58 *ppm* in order of magnitude of means from larger to smaller. The darker a unit in a component plane for a variable is, the closer the relation of this variable to the unit is. Upon investigation of the component planes, the following can be deduced:

- The most common variables are at 0.90, 1.30, 1.34 and 3.22 *ppm*, i.e. these metabolites appear to be very closely related to almost all units in the map. Especially in the case of 3.22, only samples 55 and 81 (belonging to cluster 1, eighth unit from the right at the top row) are not associated with this metabolite.
- The least common variables are 2.02 and 5.30. Very few units appear to be associated with these metabolites, with 5.30 being the least associated to the map, of the two.
- The consistent samples with regards to high intensity values in all component planes are samples 116 and 122 in cluster 3 and samples 1 and 15 in cluster 4.



**Figure 7.35:** Component planes for selected variables in the blood serum of the epilepsy patients, labelled by the chemical shift



- Samples 46 and 124 in cluster 6, and samples 3 and 58 in cluster 5, are the samples least associated with the variables.
- The four variables with the largest mean values, lying at spectral areas 1.30, 1.34, 1.58 and 0.90 *ppm* in order of appearance, are clearly more closely related to clusters 3 and 4 than any of the other clusters.
- None of the component planes is capable of describing the six clusters.

The clinical characteristics information for the six groups are given in Tables 7.22 and 7.23 below. Upon examining these summary tables, it is clear that certain groups

**Table 7.22:** Number of patients in each of the six clusters returned by SOM and the clinical characteristics *Gender*, *Seizure type* and *Response to AEDs*.

Group	Gender		Seizure type		Response to AEDs	
	Male	Female	LRE	IGE	Responder	Non-responder
1	19	6	20	5	15	10
2	13	8	16	5	10	11
3	4	1	5	0	2	3
4	15	3	11	7	7	11
5	4	11	11	4	11	4
6	10	3	12	1	7	6
<b>Totals</b>	65	32	75	22	52	45

appear to be dominated by specific characteristics. In particular, concerning the *Gender* of the patients, groups 1, 3, 4 and 6 are male-dominated, whereas group 5 is dominated by females and group 2 is balanced. Considering *Seizure type*, only group 4 is balanced whereas the rest of the groups are dominated by the LRE type, due to their larger numbers in the data. Group 5 is dominated by patients who showed improvement to AEDs treatment while all other groups are rather balanced, although group 4 contains approximately 50% more non-responders than responders. Concerning *Age*, group 4

**Table 7.23:** Number of patients in each of the six clusters returned by SOM and the clinical characteristics *Age* and *Body-Mass-Index (BMI)*.

Group	Age			BMI			
	(16-26]	(26-47]	(47-99]	(16-22]	(22-25]	(25-28]	(28-45.1]
1	7	11	7	3	9	7	6
2	6	5	10	4	3	7	7
3	0	4	1	0	0	2	3
4	11	5	2	11	5	1	1
5	7	1	7	5	7	2	1
6	0	6	7	2	3	3	5
<b>Totals</b>	31	32	34	25	27	22	23

is also the only group which is dominated by young people (16-26 years old), whereas group 3 is dominated by middle-aged patients (26-47 years old) and group 6 contains

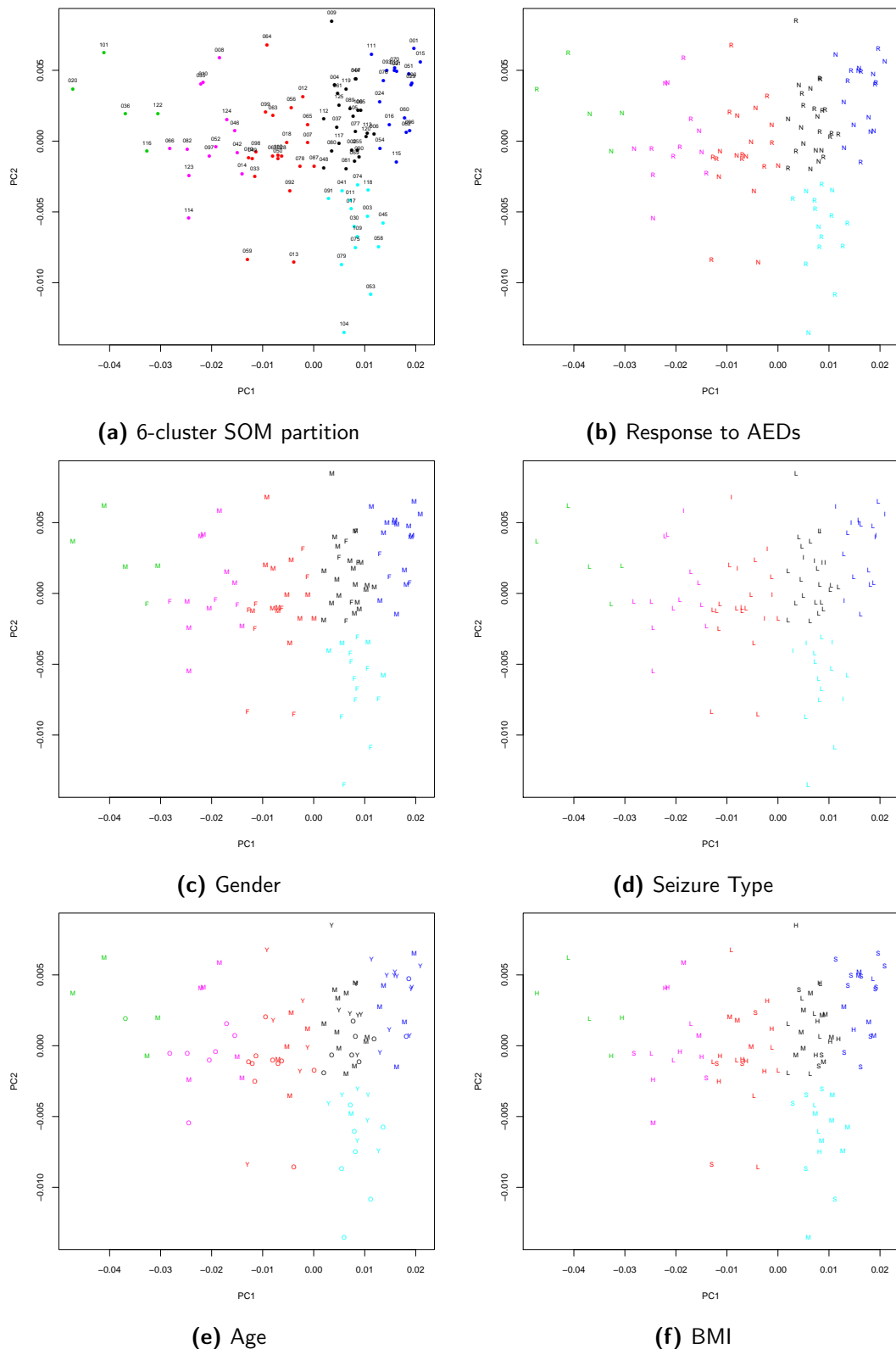
only middle-aged and old people (47-99 years old). Group 5 contains young and old patients, while groups 1 and 2 are rather balanced towards the three age categories. The information for *BMI* shows that group 4 contains mainly people with low *BMI* values (16-22], whereas groups 1, 3 contain people with higher *BMI* values (22-45.1]. Group 5 is oriented towards small to medium *BMI* values. Groups 1, 2 and 6 are rather balanced with respect to the *BMI* values.

The dominating characteristics observed in each of the six groups as derived from the SOM analyses are given below:

1. Males of all *Age* categories with LRE seizure type and improvement to AEDs treatment and *BMI* values medium to huge.
2. Males and females of all ages (although half of them are of age above 47 years) with LRE seizure type with balanced *Response to AEDs* and large to very large *BMI* values (25-45.1].
3. Middle-aged males with LRE seizure type with balanced *Response to AEDs* (although there are slightly more non-responders than responders) and large to very large *BMI* values (25-45.1].
4. Young or middle-aged males of both seizure types, non-responders to AEDs treatment and small to middle *BMI* values (16-25].
5. Young or old females with LRE seizure type, responders to AEDs treatment and small to middle *BMI* values (15-25].
6. Middle-aged and old males with LRE seizure type, balanced *Response to AEDs* treatment and rather balanced *BMI* values (although above 22 and mainly large to very large *BMI* values).

A two-dimensional projection of the epilepsy data superimposed with the clustering solution derived by the 6-cluster SOM partition (top left plot) and the five clinical characteristics can be seen in Figure 7.36. More specifically, the first two principal component scores (according to the results from Chapter 5) can be seen superimposed with the partition derived by the selected SOM clustering model. The remaining five scores plots illustrate the clinical characteristics information, with points labelled as "R" and "N" for responders and non-responders to AEDs respectively, "F" and "M" for females and males respectively, "I" and "L" for IGE and LRE seizure type respectively, "Y", "M" and "O" for the the categories of *Age*, and finally, "S", "M", "L" and "H" for the four *BMI* categories, in each of their respective plots. Figure 7.36 confirms the findings for the five clinical characteristics of the patients, with the *Age* and *BMI* score plots especially showing the discrimination of the patients into the respective categories of the characteristics. On the other hand, there is clearly no discrimination of the patients with regards to their *Seizure type* and *Response to AEDs*.

To examine whether there are any relationships among the six groups of patients and



**Figure 7.36:** Scores plots of the first two PCs, superimposed with the 6-cluster partition derived by the SOM clustering model and the information of the five clinical characteristics. The six colours correspond to the six clusters in the SOM partition of Figure 7.32, with black, red, green, blue, cyan, magenta corresponding to clusters 1-6 respectively. The labels of the points in the bottom plot correspond to the labels assigned in each of the categories of the five clinical characteristics.

their clinical characteristics, Chi-Square and KW tests will be applied. More specifically, the homogeneity of the derived 6-cluster partition with respect to the observations in each cell for the two categories of the characteristics *Response to AEDs*, *Gender* and *Seizure type* will be assessed using the  $\chi^2$  test. The remaining two clinical characteristics, *Age* and *BMI*, will be assessed using the KW non-parametric test. In this case, the null hypothesis is that the medians of the populations represented by the derived six clusters are all equal with respect to *Age* and *BMI*. The results of these tests can be seen in Table 7.24.

**Table 7.24:**  $\chi^2$  test for the homogeneity of the derived 6-clusters with respect to the proportion of observations in each of the categories of the clinical characteristics *Response to AEDs*, *Gender* and *Seizure type*, and Kruskal-Wallis test for the 6-clusters with respect to *Age* and *BMI*. The statistically significant  $p$ -values at the 95% confidence level are shown in bold.

Characteristic	P-Value	
	$\chi^2$	Kruskal-Wallis
<i>Gender</i>	<b>0.00902</b>	
<i>Seizure Type</i>	0.2983	
<i>Response to AEDs</i>	0.4158	
<i>Age</i>		<b>0.03552</b>
<i>BMI</i>		<b>0.00011</b>

Concerning *Seizure Type* and *Response to AEDs*, the  $p$ -values of the  $\chi^2$  tests, being larger than the significance level of 0.05, show that there is not enough evidence to reject the null hypothesis that the six clusters in the selected SOM partition are homogeneous with respect to these two characteristics. That is, the proportions of observations in each of the categories of *Seizure type* and *Response to AEDs* are not different in the 6-cluster SOM partition.

Considering the  $\chi^2$  result for *Gender*, the  $p$ -value of 0.00902 is clearly smaller than 0.05. Therefore, in this case, the null hypothesis is rejected, as the clusters in the 6-cluster SOM partition are not homogeneous with respect to the gender of the patients, meaning that the proportions of observations for the categories of *Gender* are different in at least one of the clusters.

The KW tests for *Age* and *BMI* return  $p$ -values of 0.03552 and 0.00011, respectively. Both values are below the significance level of 0.05, therefore there is enough evidence to reject the null hypothesis that the clusters in the 6-cluster SOM partition represent populations with equal median values. In other words, there is at least one cluster for which the median value of the population it represents is different than that of the represented populations of the rest of the clusters in the partition with respect to *Age* and *BMI*. There is clearly a relationship between these two clinical characteristics and the six clusters of the SOM partition. Table 7.23 is consistent with the findings of the KW tests. Concerning *Age* there are distinct clusters without young patients (clusters

3 and 6), or practically without middle-aged patients (cluster 5), as well as a cluster with young patients dominating (cluster 4). A similar situation is observed for *BMI*, as there are clusters dominated by patients in each of the four *BMI* categories, such as cluster 6 (patients with (28-45.1] *BMI* values), cluster 4 (patients with (16-22] *BMI* values) and clusters 1 and 5 (patients with (22-25] *BMI* values).

Thus, the SOM clustering model has been capable of discriminating the patients with respect to *Gender*, *Age* and *BMI*, but has not been successful in identifying any patterns for the patients' *Seizure type* and *Response to AEDs*.

## 7.8 Conclusions

This chapter involved the application of a number of clustering algorithms to the epilepsy data. After extensive investigation of the literature, the algorithms deemed to be the most appropriate for metabonomics data included *Hierarchical clustering*, *Optimal partitioning* with fuzzy and hard clustering methods and *Competitive learning* algorithms. The main aim was to assess the possible existence of any natural groupings in the data, and consequently identify any patterns with regards to the patients' clinical characteristics, and in particular any discrimination of the patients with respect to their *Response to AEDs*. Three non-parametric tests were applied to the results of the analyses, to assess whether there was any relationship between clinical characteristics and the partitions derived by the constructed clustering models, partitions. More specifically, the  $\chi^2$  and in some cases *Fisher's exact* tests were used with *Gender*, *Seizure type* and *Response to AEDs*, whereas the *Kruskall-Wallis rank sum* test was used with *Age* and *BMI*.

*Hierarchical* methods (HCA) involved the clustering of the data with a range of *agglomerative nesting* algorithms, the *single linkage*, the *complete linkage*, the *unweighted* and *weighted pair-group* methods using *arithmetic averages*, as well as the *unweighted* and *weighted pair-group* methods using *cluster centroids* and *Ward's* method. These algorithms cover most types of clusters from non-compact elongated (*single linkage*) to compact spherical (*Ward's* method and within-clusters sum-of-squares minimization method). Four different distance metrics were used in the construction of the agglomerative clustering models, namely the *Euclidean*, *Manhattan*, *Maximum* and *Canberra* distances. Therefore, to improve the chances of HCA identifying any natural groupings, 28 clustering models were constructed and their clustering results were compared. After extensive experimentation with the use of a range of statistics to assess the quality of fitting of the data by the clustering models (such as the *Silhouette width*, the *agglomerative coefficient* and the *cophenetic correlation*), the overall best fitting result found was to be that of the 2-cluster partition derived by the *Maximum - Ward* model. The *Silhouette coefficient* of 0.59 was the highest among all models and the cluster sizes of 63

and 34 were far more balanced compared to those of the second best model *Maximum - Average* (cluster sizes of 94 and 3).

The statistical tests, applied to the 2-6 cluster *Maximum - Ward* partitions showed that concerning *Response to AEDs* and *Seizure type*, there is no relationship with these partitions. On the other hand, the clustering models with 5- and 6-cluster partitions, were capable of discriminating the patients with respect to *Gender*. In addition, test results on the *Age* and *BMI* of the patients clearly indicated that there is a relationship between these two characteristics and the selected 2-6 cluster partitions. Therefore, hierarchical clustering models were capable of discriminating the patients with respect to their *Gender*, *Age* and *BMI*, but not with regards to their epilepsy characteristics, *Seizure type* and *Response to AEDs*.

Optimal partitioning methods, based usually on the minimisation of a cost function, were applied as a next step to HCA. More specifically, two algorithms of partitioning methods were applied to the data, a *fuzzy* clustering and a *hard* clustering algorithm.

The *fanny* fuzzy clustering algorithm described in Section 7.6.2.2 was used and 20 different fuzzy clustering models were constructed with respect to four distance metrics, *Euclidean*, *Manhattan*, *Maximum* and *SqEuclidean*, and 5 *fuzzifier* values selected after extensive experimentation in the range 1.1 – 3.0. *Silhouette coefficients* for all 20 models and for 2-6 cluster partitions confirmed that the best fuzzy clustering model was the 2-cluster fuzzy partition derived by the model with the *SqEuclidean* metric and *fuzzifier* value 2, with *Silhouette coefficient* 0.74 and clusters of size 66 and 31. Concentrating on this partition, which proved to have the largest *Silhouette coefficient* of all clustering models assessed in this work, it was confirmed by the statistical tests that as in HCA, there is a relationship only between the *Age* and *BMI* of the patients. In other words, the fuzzy clustering model was not capable of discriminating the patients with respect to their two epilepsy characteristics and *Gender*.

In the case of *hard partitioning*, the *k*-means method was the obvious choice, as it is the most popular in metabonomics data analyses. The *k*-means algorithm described in Section 7.6.5.1 was used on the epilepsy data and the optimum number of clusters was determined with the aid of a range of stopping rules, such as the *Ratkowsky-Lance* and the *Trace W* indexes, as well as the *Silhouette coefficient*. As expected from the HCA and fuzzy clustering, the 2-cluster partition derived by the *k*-means clustering model was the best hard partition. Coincidentally, it was the same partition with that derived from the best fuzzy clustering model, despite the differences in the fitting and the silhouette width values of the two models. Therefore, the selected two-cluster optimal partitioning models have the same discriminating ability with respect to the five clinical characteristics of the patients.

Finally, a category of clustering algorithms which have not been used widely in metabonomics is that of the competitive learning algorithms. The classic online *Self-*

*organizing maps* algorithm was chosen, as it has some innovative advantages compared to the other clustering methods. Apart from allowing the visualisation of the data in a map-like graph, it provides a range of visualisation tools for assessing the quality of the derived map, such as *unified distance matrix* plots, the *hit histograms*, *quality maps* and *component planes*. Two maps were chosen for comparison and analysis purposes, the main one being of size  $24 \times 2$  and a smaller one of size  $3 \times 2$ .

The available visualisation tools showed that the quality of the mapping of the data to the  $24 \times 2$  map was quite good, with the samples in all map units except three being quite close to their respective codebook vectors. The sizes of the six clusters for both maps were 25, 21, 5, 18, 15 and 13 for clusters 1-6, respectively. The six-cluster solution indicated that *ppm* areas of 5.5 - 5.1, 3.5 - 3.1, 2.1 - 2 and 1.6 - 1 are responsible for the main differences in the intensity levels of the variables in the six clusters, with the *ppm* area of 1.6 - 1 being the most important, as the variables in this area have the largest intensity values. Component planes showed that the variables at 0.90, 1.30, 1.34 and 3.22 *ppm* are very closely related to almost all map units, whereas the least common variables were at 2.02 and 5.30 *ppm*. Patients 116 and 122 in cluster 3 and samples 1 and 15 in cluster 4 were the most closely associated patients with the previously mentioned variables, while patients 46 and 124 in cluster 6 and patients 3 and 58 in cluster 5 were the least associated patients with the variables. In addition, four variables at 1.30, 1.34, 1.58 and 0.90, which were observed to have the highest mean values, were found to be clearly more closely related to clusters 3 and 4 than any other of the six clusters.

Considering the patients' clinical characteristics, clusters 1, 3, 4 and 6 were dominated by *males*, whereas only cluster 5 was dominated by *females*. Patients with *LRE Seizure Type* dominated all but one clusters, while *responders* dominated clusters 1 and 5 and *non-responders* only cluster 4. Finally, clusters 4 and 5 contained mainly patients with *small to middle BMI* values, whereas clusters 2, 3 and 6 contained patients with *large to huge BMI* values.

Statistical tests for the 6-cluster SOM partition and the five characteristics showed that the SOM algorithm was capable of distinguishing the patients with respect to their *Gender*, *Age* and *BMI*, but not their epilepsy characteristics.

Comparing the four clustering methods, it is clear that, despite the fact that none of the methods were able to discriminate the patients in terms of all clinical characteristics, two of these methods, i.e. HCA and SOM, were able to discriminate the patients also with respect to *Gender*, therefore HCA and SOM can be considered as the best overall methods for gathering information about patterns in such data regarding the clinical characteristics. An important advantage of HCA is that its results are represented in a hierarchical structure, which allows the comparison of partitions of different number of clusters, without the need to re-run the algorithm and obtain a new partition. Considering the type of clusters that have been observed in the data, which are

rather non-compact and elongated, it seems that algorithms such as *Ward's* method in HCA and *k*-means are not so suitable, since their aim is to seek compact and spherical clusters. Fuzzy clustering could have been useful if more than 2 categories of *Response to AEDs* existed in the data, as some of these categories are not clearly defined and patients in such *Response to AEDs* categories could actually lie somewhere between the two groups of responders and non-responders to AEDs. SOM has the advantage that the data can be represented in a map-like visualisation form, but such a map must contain more than 2 nodes (units) to represent the data faithfully, therefore SOM is more suitable for cases where the data contains at least 4-6 clusters. If this requirement is satisfied by the data, then SOM is the best approach with respect to the available visualisation tools for the examination not only of the patterns in the data and the quality of the analyses, but also for the investigation of any relationship between variables and map units (and consequently the patients). Overall, and considering these facts, HCA seems to be the most appropriate method for the analyses of metabonomics data of patients with epilepsy.

In general, all the clustering models derived by these various clustering algorithms, proved not to be capable of discriminating the patients with regards to their *Response to AEDs*. Further research is needed, to assess the conditions under which the unsupervised statistical techniques described in Part II of this work, could discriminate the response information with low misclassification rates. This is the aim and subject of Part III of this thesis, where, in a wide range of simulations, data sets will be generated from the epilepsy data to assess whether aspects such as the sample size of the data play any role in improving the discriminating ability of the unsupervised techniques.



# Summary

Part II of the thesis covered a range of linear and non-linear pattern recognition techniques attempting to identify any natural clustering patterns in the epilepsy data. More specifically, initially, in Chapter 5, a linear dimension-reduction technique, i.e. PCA, was applied in the data, to reduce the dimensionality of the input space of the data to two or three dimensions, making the pattern recognition procedure easier by visualising the data in a 2 or 3 dimensional representation. Results indicated that two PCs (or dimensions) are sufficient to describe 95% of the total variation of the data. Addition of the clinical characteristics information of the patients to the two-dimensional scores plots for the first two PCs confirmed that patterns were identified for the *Gender*, the *Seizure Type*, *Age* and *BMI*, but more importantly not for the *Response to AEDs*. It was also shown beyond any doubt by the use of GLM, that the patients cannot be separated with respect to their *Response to AEDs*. Loadings plots indicated the relationship between some of the variables in the data and the four clinical characteristics, as can be seen in detail in Section 5.4. In general, PCA was proved to be useful in identifying patterns for all clinical characteristics apart from the *Response to AEDs*, and for obtaining a good picture of the general structure of the data.

As PCA is used exclusively with the Euclidean distance, a non-linear dimension-reduction technique, i.e. the multidimensional scaling (MDS), was chosen for application to the epilepsy data, in the hope of deriving more information about any patterns of the patients with respect to their clinical characteristics, especially *Response to AEDs*, for which no useful information was obtained by PCA. MDS algorithms have the advantage that unlike PCA, they can be used for dimension-reduction and pattern recognition purposes with any dissimilarity (or similarity) measure. Two such methods were applied to the data, i.e. the *classical* MDS and a special case of metric MDS, i.e. *Sammon's non-linear mapping*. Four distance metrics were used for comparison purposes, with the *Euclidean* and the *Maximum* distance models giving the best results with respect to two criteria. Results of the analyses showed that both MDS methods were consistent with PCA in identifying the patterns observed in the four previously mentioned clinical characteristics, as well as in failing to identify any clustering patterns of the patients with regards to their *Response to AEDs*. Concerning the two distance metrics, in the case of *classical* MDS, the Euclidean model was slightly better in illustrating the clustering behaviour of the patients with respect to their four clinical characteristics, while in the

NLM the *maximum* had a slight edge to the *Euclidean* model, especially for the *Gender*, *Age* and *BMI* characteristics. In general, MDS was not capable of providing any further information about the clustering behaviour of the patients than PCA.

The next step in the exploratory analysis involved the application of a range of unsupervised clustering techniques to the epilepsy data, to classify if possible, the patients to groups concerning their spectral and clinical characteristics, and particularly exploring the possibility of finding any clustering behaviour of the patients with respect to their *Response to AEDs*. These clustering techniques included *hierarchical agglomerative* algorithms, *optimal partitioning* methods such as *fuzzy* and *k-means* clustering, as well as *competitive learning* algorithms with emphasis on the use of *self-organising maps*.

Concerning HCA, among many methods tested, the *Maximum - Ward* model for 2-6 cluster partitions proved to be the best at clustering the data with respect to the clinical characteristics. Specifically, using statistical tests it was shown that there was no relationship between these partitions and the clinical characteristics *Response to AEDs* and *Seizure Type*. A relationship was found between the 5-6 cluster partitions and *Gender*, as well as between all five clustering partitions and the characteristics *Age* and *BMI*. Overall, HCA was capable of discriminating the patients with respect to all their clinical characteristics, except for their *Response to AEDs*.

In the *optimal partitioning* methods, two types of *fuzzy* and *hard* clustering algorithms were applied to the data, more specifically the *fanny* and *k-means* methods respectively. In *fanny*, the 2-cluster *SqEuclidean* fuzzy clustering model was found to be the best, among a range of distance metrics and *fuzzifier* values. Results of the fuzzy analyses showed that the selected fuzzy model was consistent with the results of HCA, concerning *Age* and *BMI* of the patients, but was not capable of discriminating the patients with respect to their two epilepsy characteristics or *Gender*.

Hard clustering analyses resulted in obtaining a *k-means* clustering model, with the derived 2-cluster partition being the best. This partition proved to be the same as that obtained from *fuzzy* clustering with respect to the patients in each of the two clusters, differing only in the *silhouette width* of the patients in the two models. Therefore, the *k-means* algorithm was as successful as the *fanny* model in clustering the patients regarding their clinical characteristics.

A different clustering approach was then used, with the *self-organising maps* method being the algorithm of choice for clustering the epilepsy data. As there was no point in using a two-unit map in such an algorithm, a different approach was adopted by comparing a  $3 \times 2$  map to the maximum, recommended for the specific epilepsy data set, i.e  $24 \times 2$  SOM map. Both maps had good quality mapping of the data to their respective dimensions, especially the smaller of the two maps, giving precise information about the spectral areas of variables with high intensity values for the six clusters, and the relationship of specific variables to the patients in the six clusters. Details of this

information can be found in Section 7.8. Statistical tests confirmed that the 6-cluster SOM partition was capable of distinguishing the patients regarding their *Gender*, *Age* and *BMI* but incapable of doing the same for the two epilepsy characteristics.

Despite all previously mentioned exploratory methods being able to describe to some extent the structure of the epilepsy data, none of them was able to show any clustering structure of the data as far as the *Response to AEDs* information is concerned. The reasons behind this failure will be investigated in Chapter 8, where the PCA information will be used in an extensive number of simulations, to assess whether aspects such as the sample size of the data and characteristics of the distributions of the variables, affect in any way the discriminating ability of PCA.

## Part III

# Simulation Experiments

# Chapter 8

## Data Simulation

### 8.1 Introduction

As has already been seen in chapters 5, 6 and 7, none of the unsupervised techniques that were used to perform exploratory analysis on the epilepsy data were found to be able to discriminate the patients with respect to their response to AEDs treatment. To assess if the capability of these techniques to identify clusters of patients depends on the available epilepsy data or not, and in what way, further investigation will be needed. More specifically, two possible cases need to be assessed:

- There is no difference in the spectrum between the responders and non-responders in the epilepsy data, hence none of the techniques finds a difference.
- There is a difference, but the techniques cannot find it because either the sample size is too small or the difference is too small.

To identify which of the two cases is valid, two data sets will be generated based upon the epilepsy data with known differences in the spectra and the circumstances under which PCA can detect the groups will be examined.

In this chapter, a series of simulation studies based on the epilepsy data will be described. These involve the generation of two new data sets in each of these experiments - a reference and a test set - from the original data which includes the 97 responders and non-responders to AEDs. The original data will be called the *epilepsy data* for the simulation studies in this chapter. The simulation studies will involve the mean-shifting of various subsets of the variables in the epilepsy data - the shifted means and standard deviations of which will be used for the generation of the test set - with the purpose of comparing this set to the reference set. The size of the subsets of variables - the number of variables to be mean-shifted - has been set for five such subsets, containing 244, 120, 20, 3 and 1 variables, called *MS244*, *MS120*, *MS20*, *MS3* and *MS1* respectively from now on. These were chosen to allow for covering a wide range of variable sets between the full set of 244 variables up to a single variable where one area in the spectrum is targeted. The 3, 20 and 120 variables' subsets represent

approximately 1 %, 10 % and 50 % of the variables in the data set. Various methods of selecting which variables in these subsets to mean-shift are applied to allow comparisons of various descriptive statistics and their effect in the simulation results. These include the maximum or minimum standard deviations and the maximum mean values, called *MAXDEV*, *MINDEV* and *MAXMEAN* respectively from now on. The variables' means measure the central tendency of their distributions and those variables with the higher mean values are expected to play an important role in the discrimination of the two sets, since the information contained in most of them is described by the first two principal components at most, as it was shown in Subsection 5.3.4. The standard deviation is the most reliable statistic to measure the variability in a variable (and any potential differences among distributions of points) and by using *MAXDEV* and *MINDEV* the importance (existence or not) of the variability in the variables of the data set is assessed. It is also important to examine whether the variability of the variables is related to the distances between distributions of points, therefore standard deviation is the most suitable statistic for this purpose. The reference set will be generated in the same way from the epilepsy data but without mean-shifting. Sample sizes of 100, 500 and 1000 were chosen to represent a good range of small to large samples (Osborne and Costello, 2004), with the case of 100 samples being quite close to the sample size of the original epilepsy data. This range is also suggested by previous research (Guadagnoli and Velicer, 1988). These sample sizes will be called *S100*, *S500* and *S1000* respectively in this chapter. A very high level of discrimination will be used (almost complete separation for a 1% misclassification rate) to increase the probability of uncovering the structure of the data in all cases.

Linear discriminant analysis (LDA) will be applied, for the first two PCs only, to the PCA scores plot of each experiment based upon the generated data, to obtain a linear boundary between the two sets and to allow the estimation of offset values when the PC analysis discriminates between the two sets with misclassification error rates of 20%, 15%, 10%, 5% and 1%. These threshold values were chosen after extensive experimentation, so that they cover a wide range of error rates (Rubingh et al., 2006). An introduction to *supervised classification* including a description of *two-class classifiers*, the *Bayes decision rule* and of *discriminant functions*, as well as a description of *linear discriminant functions* which will be used in the simulation experiments, is given in the next section (8.2).

In Section 8.3, a detailed description of the steps followed to perform these simulation studies is given, followed by the results of the simulations in each of the pre-selected cases. From these, conclusions will be reached on the identification of thresholds for the offsets, the number of samples in the two data sets and the number of mean-shifted variables which will allow PCA to discriminate between the two artificial sets. In general, if small changes in the test set, compared to the reference set, reveal the

existence of structure in the results of an unsupervised technique, e.g. PCA, then we would conclude that the epilepsy data analysed in Chapters 5-7 have no structure, whereas ideally the results should show that only a larger sample is required.

## 8.2 Supervised Learning Techniques

### 8.2.1 Introduction

An important stage in a pattern recognition problem is the application of *classification* techniques to investigate the structure of the data. These techniques are divided into two categories, *unsupervised* (or clustering) techniques and *supervised* classification techniques. Clustering techniques were described and applied in the epilepsy (unlabelled) data in Chapter 7, with the purpose of exploring the data structure for potential groups and the features which distinguish these groups from each other. In this chapter, supervised classification techniques will be described and applied to the simulated data to discriminate between the two artificially generated data sets. In supervised techniques, a learning set of multivariate observations is given, with each observation labelled as belonging to one of  $C$  predefined classes of similar characteristics. If each of the observations is assigned a unique class label, then the observations are described as *labelled* observations. A *classifier* combines the input variables in such a set to define a discrimination rule for classification purposes. There are two main purposes of applying supervised techniques to data (Izenman, 2008):

**Discrimination** To construct a classifier using the information in a learning or training set of labelled observations (i.e. each observation has been assigned to one of the classes) such that it will separate the predefined classes as much as possible.

**Classification** To use the constructed classifier to predict the class of new unlabelled observations.

### 8.2.2 Supervised Classification

In general, if  $\mathbf{x}$  is a vector containing a set of measurements, classification involves the assignment of this vector to one of  $C$  classes,  $c_i, i = 1, \dots, C$ . To achieve this, the measurement space needs to be partitioned into  $C$  regions  $r_i, i = 1, \dots, C$ . This can be done by using a *decision rule*. Thus, if a vector  $\mathbf{x}$  is in the region  $r_i$ , it belongs to class  $c_i$ . The boundaries between the regions  $r_i$  are called the *decision* boundaries or *decision surfaces* (Webb, 2002). Each such region is not necessarily convex, as it may consist of many disjoint regions.

### 8.2.3 Two Class Classifiers

*Two class* or *binary* classifiers are used when decisions have to be made to classify a sample to one of two predefined groups. In the simulation experiments, the two groups predefined by the simulation algorithm are the regions of feature space occupied by the reference and the test data set. Therefore, a binary classifier (and one discriminant function and decision boundary) will be used to investigate whether a sample in the data space belongs to the reference or the test data set. Visually, this is accomplished by drawing a border between the two data sets of samples, e.g. in a scores plot, with the samples from each data set lying mainly on one of the sides of the border. Due to the way the two data sets are generated, their variance structure is similar. That means that it is possible to achieve linear separation of the two sets and there is no need for highly complex boundaries. Therefore, a linear classifier could suffice to discriminate between the two data sets.

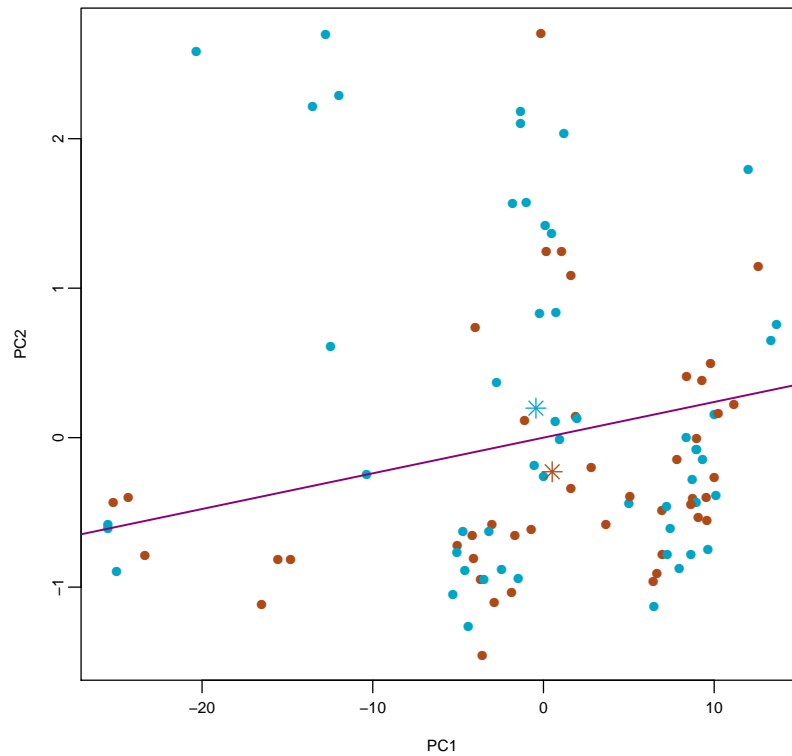
There are many binary classifiers which can be used in metabonomics data. The most popular include the *Euclidean Distance to Centroids* (EDC), *Linear Discriminant Analysis* (LDA) and *Partial Least Squares Discriminant Analysis* (PLS-DA), *Learning Vector Quantisation* (LVQ) and *Support Vector Machines* (SVM) (Breerton, 2009; Webb, 2002). Classifiers such as PLS-DA, LVQ and especially SVMs are more powerful than EDC or LDA in obtaining boundaries between classes but can be very complex and computationally intensive. These are usually the preferred choice of classifier for classification problems where boundaries of very high complexity are required. In the case of the simulation experiment, it is sufficient to use either EDC or LDA. Since LDA takes into consideration the different variances of each variable and the correlation between the variables (so that correlated variables are not weighted too much against other variables measuring different properties (Breerton, 2009)), whereas EDC does not, LDA was the classifier of choice in these simulation experiments. An example of the application of the LDA classifier for the construction of a linear decision boundary in the epilepsy data for the two groups of patients corresponding to responders and non-responders to AEDs can be seen in Figure 8.1. In this case, it is clear that the two groups (responders and non-responders to AEDs) are not separable in the first two PCs, as there are more than 30 misclassified patients in both groups. The range of values of PC1 is far greater than that of PC2 which might have affected the classification of the patients.

### 8.2.4 Bayes Decision Rule

If

$$p(\mathbf{x} \in c_i) = p(c_i), \quad i = 1, \dots, C$$





**Figure 8.1:** Illustration of an LDA decision boundary for the original epilepsy data. The data is log-transformed. The two groups are the responders (brown points) and non-responders (blue points) to AEDs from the 97 patients in the original epilepsy data. The stars represent the group means estimated using the first two principal components of the epilepsy data.

is the prior probability that a randomly selected vector  $\mathbf{x}$  belongs to class  $c_i$ , and

$$p(\mathbf{x}|\mathbf{x} \in c_i) = p(\mathbf{x}|c_i), \quad i = 1, \dots, C$$

is the class-conditional density function of  $\mathbf{x}$  for class  $c_i$ , then the posterior probability (the probability of belonging to class  $c_i$  given the observation vector is  $\mathbf{x}$ ) is given by *Bayes's Theorem* as (Izenman, 2008)

$$p(\mathbf{x} \in c_i|\mathbf{x}) = p(c_i|\mathbf{x}) = \frac{p(\mathbf{x}|c_i)p(c_i)}{p(\mathbf{x})} \quad (8.2.1)$$

where  $p(\mathbf{x})$  is the unconditional probability density function of the vector  $\mathbf{x}$ . A decision rule can then be to assign vector  $\mathbf{x}$  to the class with the higher posterior probability. That is, if

$$p(c_i|\mathbf{x}) > p(c_j|\mathbf{x}), \quad j = 1, \dots, C, \quad j \neq i \quad (8.2.2)$$

then assign  $\mathbf{x}$  to class  $c_i$  (Webb, 2002). Using 8.2.1 in 8.2.2, the rule becomes

$$p(\mathbf{x}|c_i)p(c_i) > p(\mathbf{x}|c_j)p(c_j), \quad j = 1, \dots, C, \quad j \neq i \quad (8.2.3)$$

as the unconditional probability density  $p(\mathbf{x})$  is independent of the class and does not affect the classification decision (Bishop, 1997). Equation 8.2.3 is known as the *Bayes rule for minimum error*. In the case of  $C = 2$ , 8.2.3 becomes such that  $x$  is assigned to class  $c_1$  if

$$\frac{p(\mathbf{x}|c_1)}{p(\mathbf{x}|c_2)} > \frac{p(c_2)}{p(c_1)} \quad (8.2.4)$$

and otherwise  $x$  is assigned to class  $c_2$ .

## 8.2.5 Discriminant Functions

The Bayes decision rule is based on knowledge of the class-conditional density functions,  $p(\mathbf{x}|c_i)$ . Another approach for obtaining a classification rule is by defining a discriminant function. A discriminant function is a function of an observed vector  $\mathbf{x}$  which provides a classification rule. If  $f_i(\mathbf{x})$  for  $i = 1, \dots, C$  is a set of discriminant functions, then the observed vector  $\mathbf{x}$  is assigned to class  $c_i$  if

$$f_i(\mathbf{x}) > f_j(\mathbf{x}), \quad j = 1, \dots, C, \quad j \neq i .$$

The regions where the discriminant functions are equal determine the decision boundaries, so that for two contiguous regions,  $r_i$  and  $r_j$ , the decision boundary which separates them is given by

$$f_i(\mathbf{x}) = f_j(\mathbf{x}) .$$

In the case of  $C = 2$ , a single discriminant function is used of the form

$$f(\mathbf{x}) = f_1(\mathbf{x}) - f_2(\mathbf{x})$$

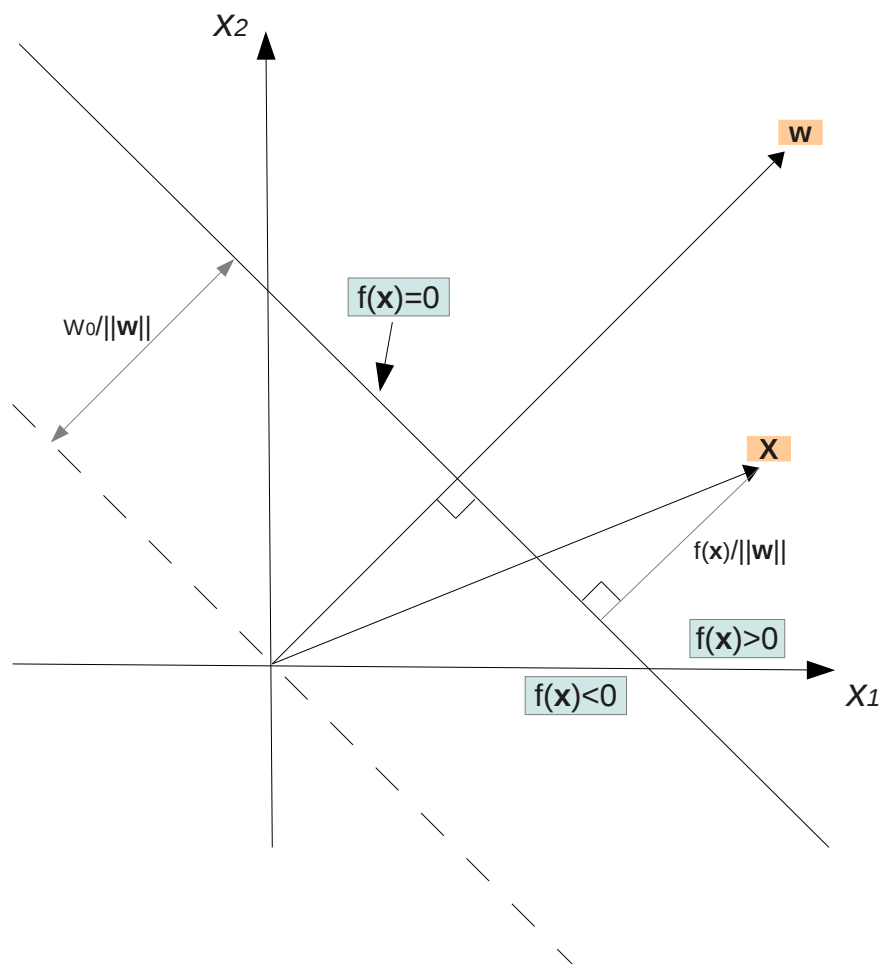
and the decision rule can be considered as assigning the vector  $\mathbf{x}$  to class  $c_1$  if  $f(\mathbf{x}) > 0$  and to class  $c_2$  if  $f(\mathbf{x}) < 0$  (Bishop, 1997). The main difference between the Bayesian decision and discriminant function approaches is that the form of the discriminant function to use is not imposed by the assumed distribution of the data. It may depend either on knowledge about the observed vectors, or the functions' parameters can be adjusted by training procedures (Webb, 2002). For the purposes of the simulation experiments, the use of linear discriminant functions has been employed.

### 8.2.5.1 Linear Discriminant Functions

Linear discriminant functions are the simplest forms of discriminant functions, as they are linear combinations of the components of a vector  $\mathbf{x} = (x_1, \dots, x_n)^T$ . Therefore, a linear discriminant function can be written as

$$f(\mathbf{x}) = \sum_{i=1}^n w_i x_i + w_0 = \mathbf{w}^T \mathbf{x} + w_0 \quad (8.2.5)$$

which is fully specified by the weight vector  $\mathbf{w}$  and the threshold weight  $w_0$  (Webb, 2002). Geometrically, equation 8.2.5 is a hyperplane with orientation in the direction of  $\mathbf{w}$  and perpendicular distance  $\frac{w_0}{\|\mathbf{w}\|}$  from the origin (Bishop, 1997). The value of  $f(\mathbf{x})$  for a vector  $\mathbf{x}$  is the perpendicular distance of  $\mathbf{x}$  from the hyperplane, as can be seen in Figure 8.2 (based on (Webb, 2002), p. 20 and (Bishop, 1997), p. 79).



**Figure 8.2:** Geometry of a linear discriminant function in a 2-dimensional input space  $(x_1, x_2)$ .

## 8.3 Simulation Procedure

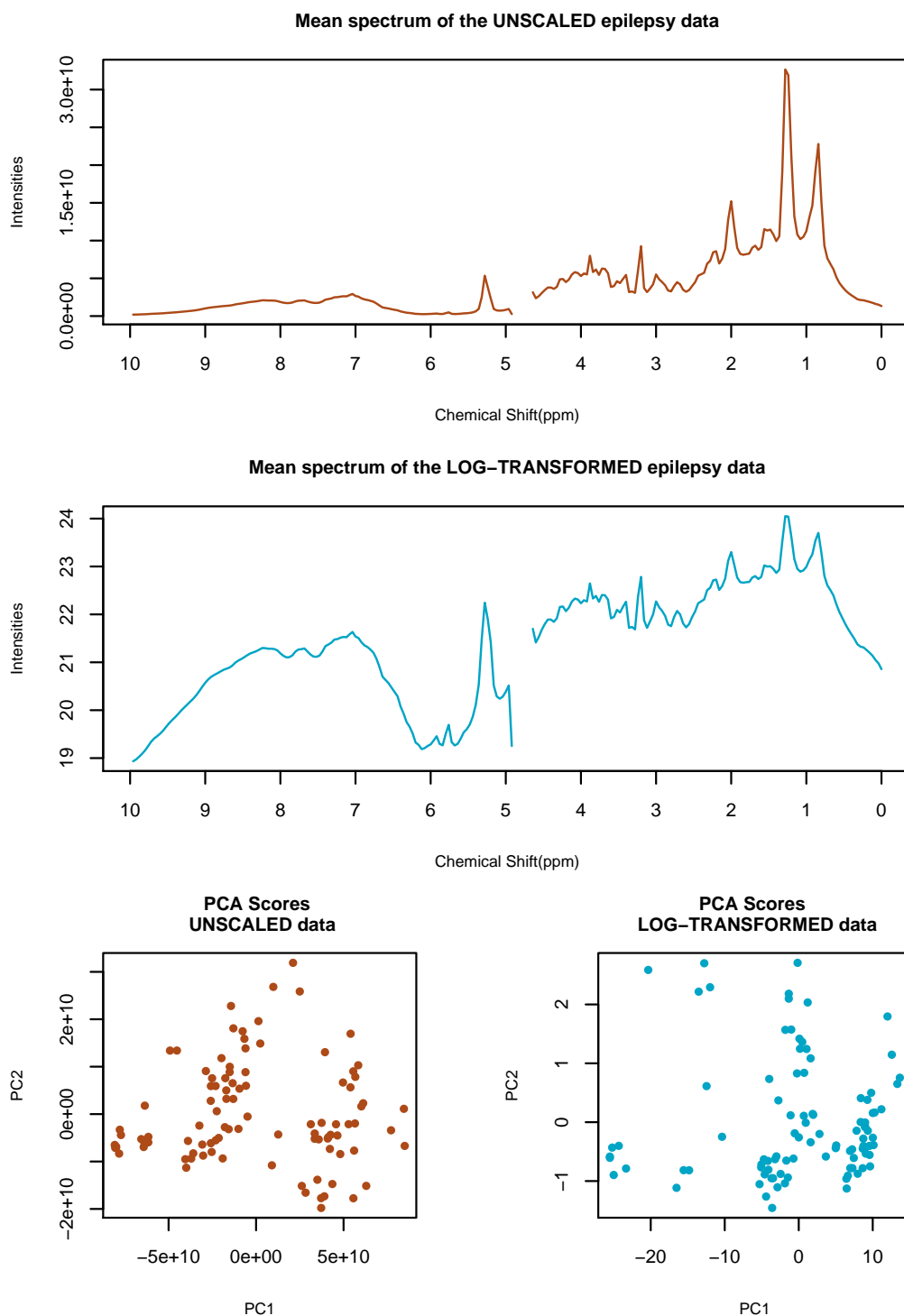
In this section the various steps of the simulation algorithm are described in detail. The algorithm has been developed in the script language of R, the free software and open source environment for statistical computing and graphics.

### 8.3.1 Preparation of the Epilepsy Data

The original epilepsy data consists of 97 patients who are the responders and non-responders to AEDs treatment. As is usual in proton NMR metabonomic data, the recorded NMR spectra include resonances which do not correspond to any endogenous metabolites (Ross *et al.*, 2007). Spectral regions which do not contain any endogenous metabolites are not useful in data analysis and therefore need to be removed before any data analysis is performed. Such regions are the spectral ranges below 0 *ppm* and above 10 *ppm* which do not contain any endogenous metabolites (Williams and Fleming, 1995). In addition, the spectrum resonances in the spectral range 4.7 – 4.9 *ppm*, which are the remaining water resonances after the application of water suppression techniques in the spectra, need to be excluded as well, since the analysis of signals of metabolites below the water resonances is not possible as the water peak dominates the proton NMR spectrum, affecting the multivariate data analysis of the spectral peaks of interest (Ross *et al.*, 2007). After the exclusion of these regions from the spectral data, the remaining spectral data, that will be used to generate the two artificial data sets for the simulation experiments, consist of 97 subjects and 244 variables in the spectral range 0.02 – 9.98 *ppm*. The elements of the data matrix were log-transformed (natural log) to reduce the influence of large intensities of outliers and large peaks in the spectra - such as the large peaks in the epilepsy data in the range 1 – 2 *ppm* - and to increase the symmetry of the distributions of intensities (Brereton, 2009). The effect of the log-transformation can be seen in Figure 8.3 compared to the raw data, both in the mean spectra and the 2D PCA scores plots. From the comparison of the mean spectra, it is clear that in the log-transformed data, the very large peaks in spectral regions such as the region 1 – 2 *ppm* do not dominate the spectral data any more. The code developed for the preparation of the epilepsy data can be seen in function `createDataClass()` on page 311.

### 8.3.2 Generation of the Reference Data Set

The simulation experiments aim to investigate the discriminating ability of the PCA unsupervised technique for metabonomic data of this type by comparing - using appropriate statistical analyses - pairs of artificially generated data sets, using various parameters such as the number of variables and method of selecting the variables to



**Figure 8.3:** Illustration of the effect of log-transforming the epilepsy data. The UNSCALED data are compared to the LOG-TRANSFORMED data with respect to their mean spectra and PCA scores plots. In the PCA function the data are mean-centred. Both the UNSCALED and the LOG-TRANSFORMED data are not row-scaled.

mean-shift, and the samples sizes. To achieve this, the parameters of the multivariate distribution of the epilepsy data obtained after the preparation steps described in Subsection 8.3.1, will be used in the simulation of the pairs of data sets. A reference data set will be generated in each simulation using a random multivariate normal distribution generator with mean as the mean vector and dispersion matrix as the covariance matrix of the multivariate distribution (spectral data) obtained as described in Subsection 8.3.1 from the epilepsy data.

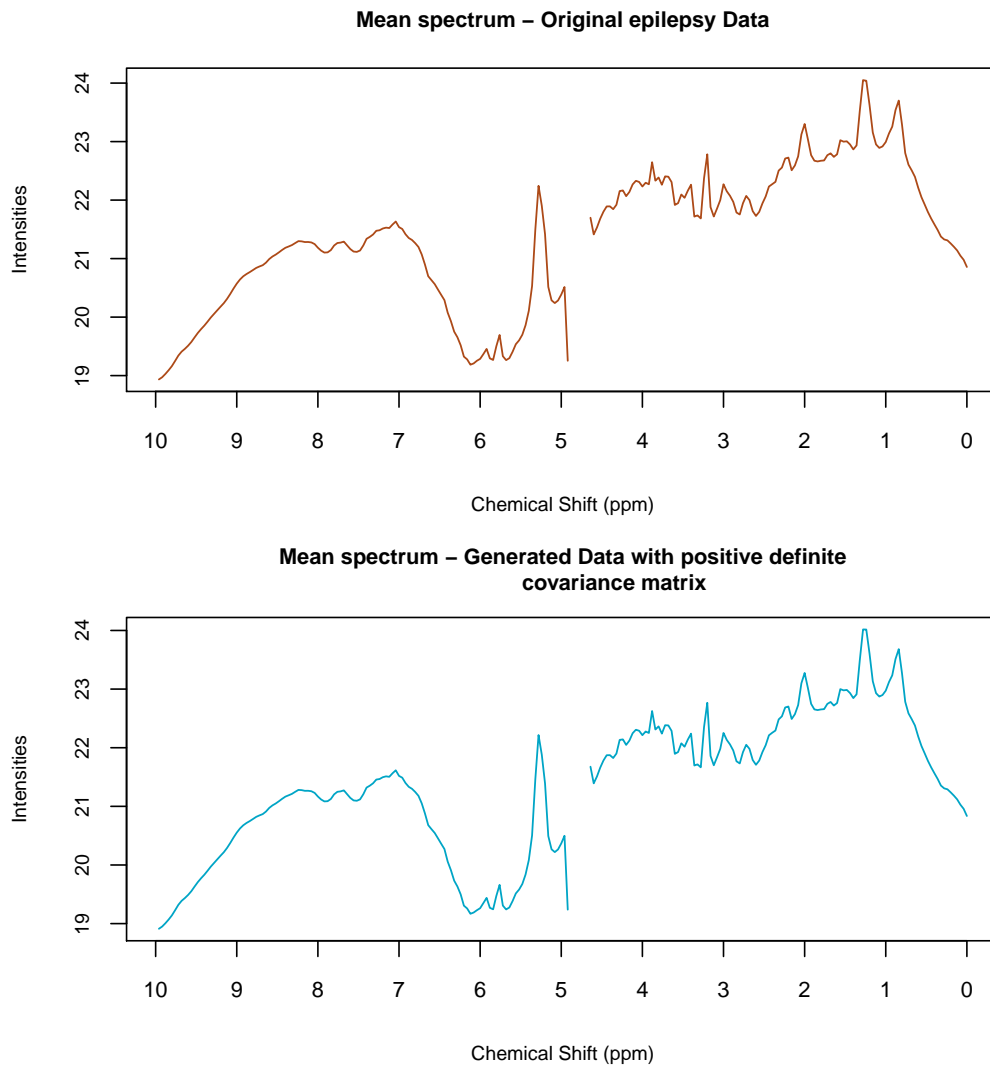
However, because the original data matrix is of dimension  $97 \times 244$ , its covariance matrix is not positive definite. As the R function `mvrnorm()`, used to generate the artificial data, applies eigen-decomposition of the covariance matrix, this cannot be singular, therefore first it is necessary to convert the matrix to positive definite. The covariance matrix of the data was converted using the R function `make.positive.definite()` from package `corpcor`. In the conversion, after experimentation and investigation, a tolerance level of  $5 \times 10^{-6}$  has been chosen, to ensure that all eigenvalues of the new covariance matrix are positive. Figure 8.4 illustrates the fact that using the positive definite version of the covariance matrix does not affect the structure of the data, which remains similar to that of the original epilepsy data. The generated reference data set  $\mathbf{X}$  follows a multivariate normal distribution of the form

$$\mathbf{X} = \mu + \mathbf{SZ} \sim N(\mu_e, \Sigma_e)$$

where  $\mathbf{Z}$  is a vector of independent standard normal deviates,  $\mu_e$  is the vector of means of the epilepsy variables and  $\Sigma_e$  the covariance matrix of the epilepsy variables with  $\Sigma_e = SS^T$  (Ripley, 1987). Alternatively, the vector of standard deviations of the epilepsy variables could be used but it is not preferable as in this case no information is retained in the generated data sets about the covariances of the variables in the epilepsy data, therefore the distribution of the generated data will not be as close to that of the original epilepsy data as it would be in the case of using the covariance matrix. Function `generateSet()` which contains the R code developed for the generation of a reference data set, can be seen on page 314.

### 8.3.3 Generation of a Test Data Set

The procedure for generating a test data set is similar to that used in the generation of the reference data set. In this case though, mean-shifting of a pre-selected set of variables takes place before the generation of the data. In each simulation experiment the number of variables to mean-shift is selected (referred to as cases MS244, MS120, MS20, MS3 or MS1), a statistic (MAXDEV, MINDEV or MAXMEAN) is applied to the epilepsy data to identify the variables to mean-shift and an offset chosen after extensive experimentation is added to the means of the selected variables (see Section 8.1 for the



**Figure 8.4:** Comparison of the original epilepsy mean spectrum (brown) to the generated mean spectrum (blue) using the positive definite covariance matrix obtained from the epilepsy data. As the original data contain 97 samples, the generated data also contains 97 samples for easier comparison. In both plots the data are log-transformed but not row-scaled.

definition of the settings used in the simulation experiments). The number of samples in the test data set is the same as in the reference data set. The generated test data set  $\mathbf{X}$  follows a multivariate normal distribution of the form

$$\mathbf{X} = (\boldsymbol{\mu} + \text{offset}) + S\mathbf{Z} \sim N((\boldsymbol{\mu}_e + \text{offset}), \Sigma_e)$$

where  $\mathbf{Z}$  is a vector of independent standard normal deviates,  $\boldsymbol{\mu}_e$  is the vector of means of the epilepsy variables and  $\Sigma_e$  the covariance matrix of the epilepsy variables with  $\Sigma_e = SS^T$  as in the reference data set case (Ripley, 1987). The `offset` term is the vector of logarithmic values added to the vector of means of the variables pre-selected for mean-shifting. For example, adding an offset of 1 (on the logarithmic scale) to the

mean of a variable is equivalent to increasing the variable's mean approximately three-fold on the original scale ( $\approx 2.72$  times). Function `generateSet()` which contains the R code developed for the generation of a test data set, can be seen on page 314. Table 8.1 illustrates all the simulation experiments done with the offset ranges used initially (these were refined later on).

**Table 8.1:** List of simulation experiments. The sample sizes refer to both the reference and the test data set and the two data sets have equal sample sizes in all experiments. The offsets represent the multiplicative factors on the original scale of the data.

Subset of variables	Offset Range	Sample Sizes
MS244	2.0 - 20.5	
MS120	1.0 - 2.0	
MS20	1.0 - 2.5	S100, S500, S1000
MS3	2.0 - 8.0	
MS1	4.0 - 50.0	

### 8.3.4 Row-scaling of Data Sets

In each simulation experiment, the samples of the two artificially generated data sets are combined to one data set, to allow for statistical analyses to be applied to the generated data. An important consideration before applying any statistical analysis to the data is to decide if it will be of benefit to apply row-scaling of the data to a constant total to make the spectra more comparable. This scaling is applied to each and every sample in the data set. Each variable (column of the data matrix) is divided by the sum of all variables in the sample, effectively replacing element  $x_{ij}$  by

$$\frac{x_{ij}}{\sum_{j=1}^J x_{ij}}$$

in sample  $i$  and variable  $j$ , where  $J$  is the number of variables. After this scaling operation,

$$\sum_{j=1}^J x_{ij} = 1$$

for each sample in the data set, meaning that the absolute intensities have become proportions. Although the log-transformation of the epilepsy data reduced the influence of some very high peaks in the spectra, there are still some high peaks, as seen in Figure 8.3, which, upon row-scaling the data, might affect the samples in a negative way. That is, variables with very high values in most of the samples in the data are not necessarily relevant to the pattern recognition stage of the simulation but can still cause significant reduction of the variance of more relevant variables with smaller values (Brereton, 2009). Therefore, caution is needed when applying row-scaling in multivariate data.



It should be noted that there is no point in applying row-scaling in case MS244. Figure 8.5 illustrates why row-scaling should not be applied to the case MS244. The selected offsets of 2.83, 4.71 and 15.96 correspond to misclassification rates of 20%, 10% and 1% respectively, covering thus the whole range of possible offsets. From the mean spectra plots in Figure 8.5, it is clear that in the row-scaled data discrimination of the two data sets is not possible, whereas in the raw data the two data sets are separated with only a 10% error rate. The PCA scores plots indicate that the two data sets are separated far more clearly in the original data case than in the row-scaled case. Therefore, row-scaling to a constant total will not be applied in case MS244.

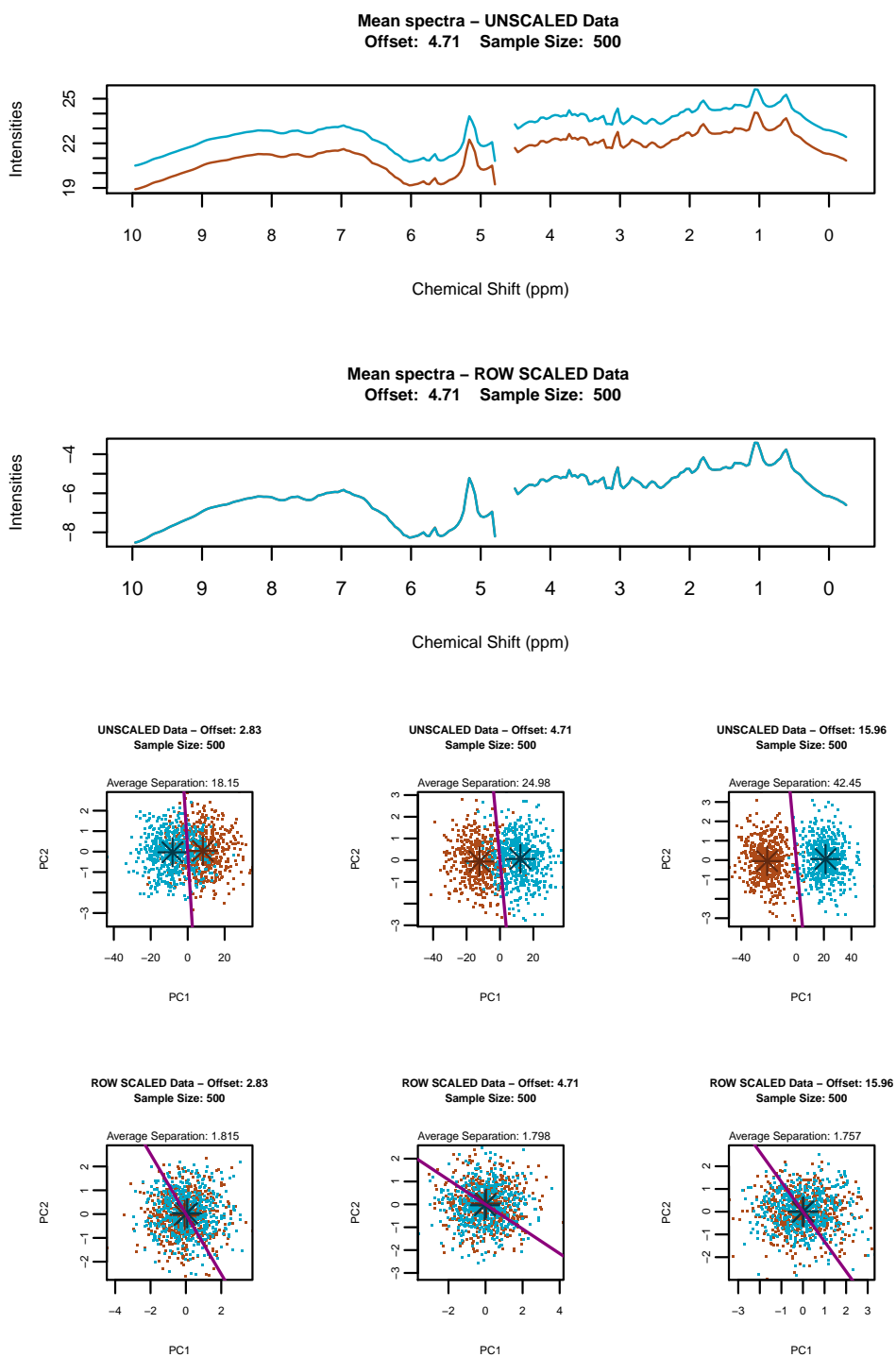
Figure 8.6 illustrates the effect of applying row-scaling to the original epilepsy data. It is clear, that by row-scaling the data the intensity levels of the samples in the data have been adjusted so that differences between them are much smaller. Hence, the effect of samples with generally very high intensity values compared to those samples with small intensity values in the analyses will be far smaller than in the original raw data. However, row-scaling was not used initially in the experiments done, as the results obtained from the unscaled data compared (in the conclusions of this chapter) to those obtained from the row-scaled data were found not to differ noticeably. Row-scaling of the epilepsy data set is accomplished using the function `createDataClass()` on page 311.

### 8.3.5 Column-scaling of Data Sets

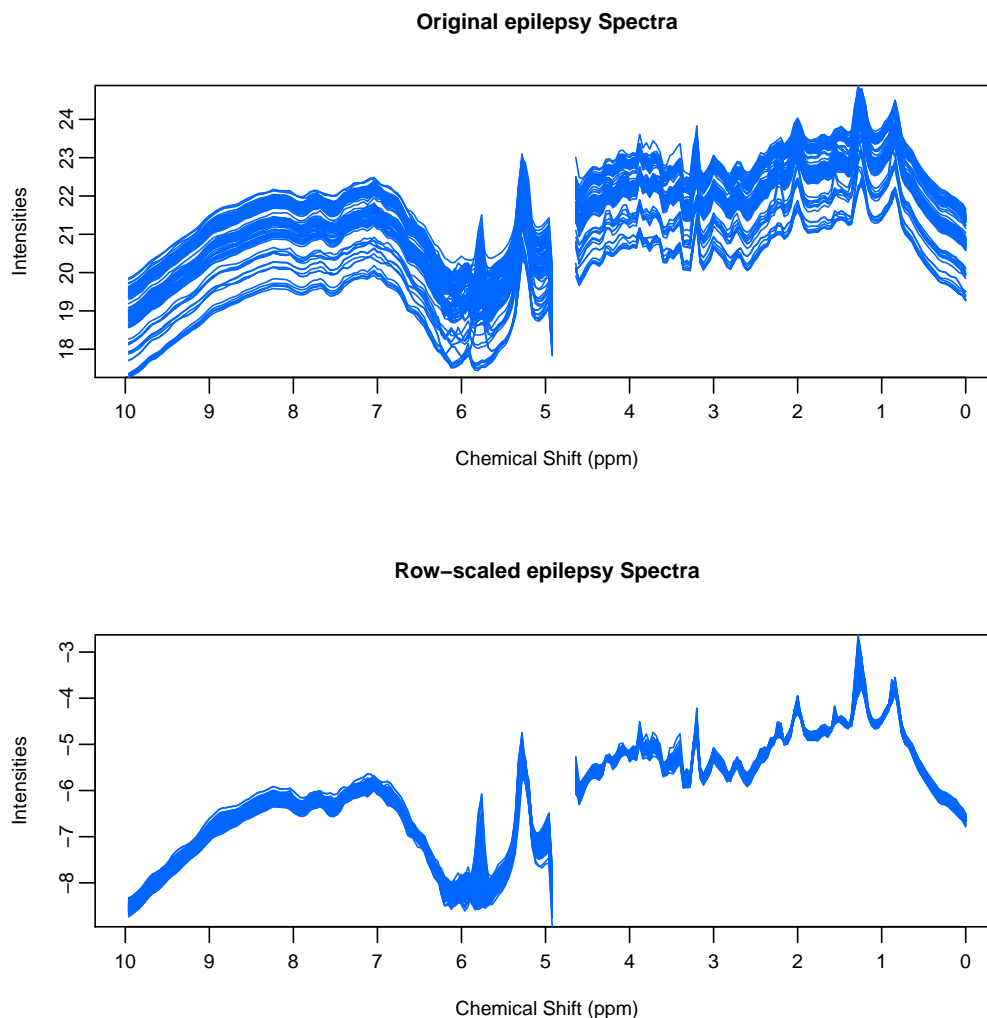
The row-scaled epilepsy spectra in Figure 8.6 indicate that there are still variables which are very intense in magnitude (in the range  $2 - 0.8$  ppm) which could dominate the analysis and variation in smaller variables might not influence the results as much as it probably could. Therefore, we need to ensure that all variables have similar influence in the analysis of the data. Column-scaling takes care of this problem. Figure 8.7 illustrates the effect of column-scaling on the original row-scaled epilepsy data. As described in Subsection 4.4.2, mean-centring is the usual column-scaling method used in multivariate data. Before applying column-scaling, the two generated data sets are joined to create a new data set which contains all samples from the two data sets. For the two groups (sets) in the generated data, it is preferable to calculate a global mean for each variable (instead of using an overall mean as in mean-centring), using weighted centring for two groups with the general formula for weighted centring becoming for  $N_c = 2$

$$\bar{x}_j = \frac{\bar{x}_{(ref,j)} + \bar{x}_{(test,j)}}{2}$$

where  $\bar{x}_{(ref,j)}$  and  $\bar{x}_{(test,j)}$  are the column means of variable  $j$  in the reference and test data sets respectively. The use of a global mean as described is recommended in multivariate analyses when the data contains more than one group (as in our case),



**Figure 8.5:** Illustration of the mean spectra of the UNSCALED and the ROW-SCALED data sets for the case MS244. The blue colour represents the test data set and the brown the reference data set. The row-scaling was applied before log-transformation took place. Both UNSCALED and ROW-SCALED data are log-transformed.

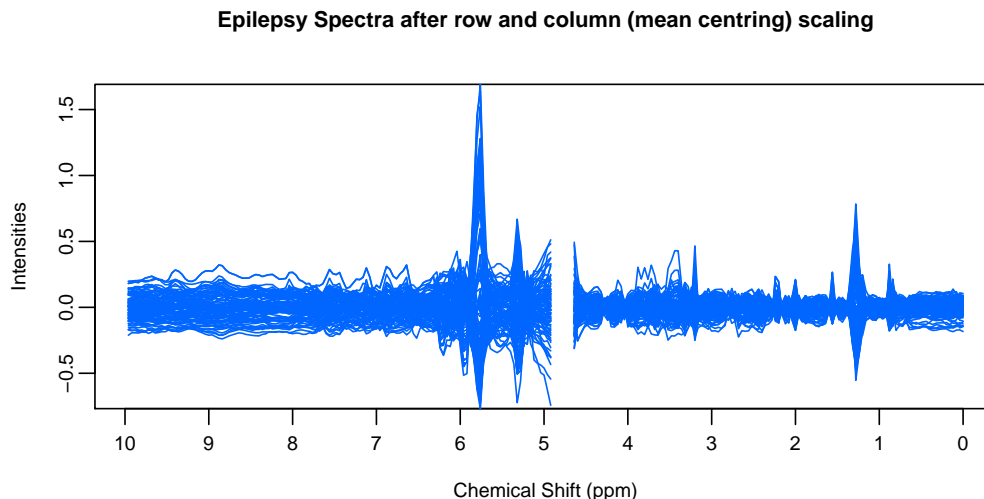


**Figure 8.6:** The original epilepsy spectra before (top) and after (bottom) row-scaling. In both plots the data are log-transformed after any scaling.

since the global mean is not biased towards any of the groups in the data (Breton, 2009). This does not affect the outcome of the simulation experiments in our case, as the two groups have the same sample size in all experiments. Column-scaling of the epilepsy data set is accomplished using the function `simulateData()` on page 318.

### 8.3.6 Principal Component Analysis

PCA is applied to the data set obtained after applying column-scaling as described above, to investigate whether the PCA technique (or any clustering technique for that matter) can discriminate between the two data sets, with respect to the selected parameters in each simulation experiment. A statistic called the *average separation* (between two samples of points) will be used to estimate the distance between the two data sets



**Figure 8.7:** The row and column (mean-centred) scaled epilepsy spectra. The plot is log-transformed after being row-scaled and before being column-scaled.

from the PCA scores (Webb, 2002). This measure is defined as the average distance between all pairs of points, with one point in each pair coming from each sample. If the sample size for the reference set data is  $n_r$  and of the test data set  $n_t$ , then the *average separation* between the reference and the test data set is given by

$$D_{avsep}(r, t) = \frac{1}{n_r n_t} \sum_{i=1}^{n_r} \sum_{j=1}^{n_t} d(x_i, y_j)$$

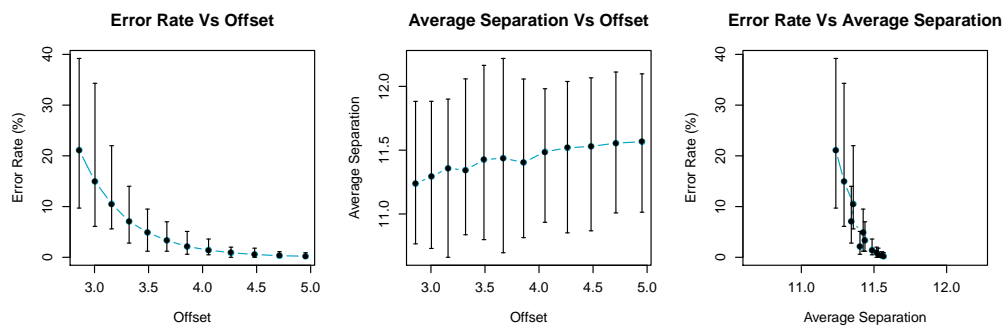
where  $x_i$  and  $y_j$  are the  $i^{th}$  and  $j^{th}$  points in the reference and test data set respectively, and  $d$  is a distance metric between  $x_i$  and  $y_j$ . In these simulation studies the Euclidean distance metric was used, as it is the most commonly used one in such studies. The *average separation* is estimated using the Euclidean distance matrix of the PCA scores of the first two components. The centre points plotted in the PCA scores plot (in Figure 8.10 and all such figures) are estimated as the mean values of the PCA scores for the two data sets. According to Webb (2002), although there are many measures of distance between distributions, only the *average separation* measure is of practical interest, as the use of other measures requires numerical integration and estimation of the probability density functions from samples. Function `simulateData()` developed to perform PCA can be seen on page 318.

### 8.3.6.1 PCA Scores and *Average Separation* Plots

A series of scores plots is used to assess the capability of PCA to discriminate between the two data sets. This involved experiments with offsets in the range 1.0 – 50 (0 – 3.91 on the log scale), as seen in Table 8.1. Depending on the experiment parameters, the

average separation value indicated a distance threshold necessary for the two data sets to be linearly separated. To explore the behaviour of the two distributions of points using the various parameters, 100-run experiments were executed and the average separation values versus offsets in 100 runs were plotted. These plots allow conclusions about the required offsets to achieve LDA misclassification rates of 1 – 20%, which consequently will indicate when PCA can discriminate between the two data sets.

The information concerning the *average separation* (value and mean in each run) and plots of this, was taken after 100 runs of the simulation algorithm and the mean values of this statistic are used in the plots. The average separation versus offsets plot is superimposed with vertical error bars, such that the top and bottom of a bar correspond to the maximum and minimum average separation at the respective offset point in 100 runs of an experiment. The error bars are used to show the stability of the *average separation* at each offset in 100 runs of each experiment. The larger the range of the *average separation* (the width of the error bar variation) in an offset, the less stable the statistic is at this offset. An example of this fact can be seen in Figure 8.8. In Figure 8.8,



**Figure 8.8:** Example of a statistics versus offsets plot with error bars superimposed.

the range of values (width of the error bar variation) of average separation in 100 runs, at offset 3.67 is clearly larger than that at offset 4.06. Therefore, in this case, the statistic is slightly more stable at offset 4.06 than offset 3.67. Function `plotSimStats()` on page 328 creates the average separation plots. As the PC scores were plotted in single run experiments, there might be some inconsistency in the calculated values of the average separation in an MS case between the three offsets. For instance, in MINDEV, the average separation values for 20 % and 10 % misclassification rate are 11.59 and 11.12 respectively in MS120 (Figure B.1), but the statistics' plot in Figure B.2 in 100 runs shows clearly that the average separation increases as the misclassification rate decreases (offset increases).

To allow the comparison of the values of the two statistics, (*misclassification rate* and *average separation*), as they are measured on completely different scales, the *coefficient of variation* (CV), will be used. This statistic is computed as the ratio of the standard deviation to the mean, multiplied by 100. The smaller the value of CV of a variable, the less dispersed the variable is. Due to its formula of computation, CV is unitless,

therefore, it describes the dispersion of a variable in a way which is independent of the variable's unit of measurement. It allows comparison of variables of different scales, which is not possible with dispersion statistics such as the standard deviation. The only requirements for its application are that the mean of a variable must not be zero and the variable contains only positive values. CV will be applied in each offset for the 100 runs of the simulation algorithm for both statistics. More specifically, it will be applied to the values described by the error bars in the statistics plots versus the offsets. Comparison of the values for the two statistics should indicate which of the two is more affected by aspects of the simulation such as offsets and sample sizes, and which of the two statistics is more stable. Function `computeCV()`, which has been developed to compute the CV for both statistics, can be seen on page 330.

### 8.3.7 Linear Discriminant Analysis (LDA)

An important step in the algorithm is to apply LDA to the first two PCs of the combined simulated data set to assess the effects of the variables' mean-shifting and of the change of sample sizes in the two data sets with regard to the *misclassification rate*. More specifically, LDA is applied to the first two PCs of the data in each case, and the misclassification rate is calculated for each specific offset. In addition, LDA produces boundaries for the two data sets in the scores plots. Similarly to the average separation versus offset plots, misclassification error versus offset plots are drawn using the average error rates in 100 runs of the algorithm, superimposed with vertical error bars, corresponding to the maximum and minimum misclassification rate at that offset point in 100 runs of the experiment. The error bars for the *misclassification rate* are similar to those of the *average separation*, as described in Subsection 8.3.6.1. However, the width of the error bar variation in the misclassification error versus offset plots is greater than in the average separation versus offset plots, therefore the *misclassification error* depends far more on the offsets than the *average separation*. Function `plotSimStats()`, on page 328, creates the misclassification rate plots.

### 8.3.8 Simulation Algorithm

The simulation procedure can be summarized in the following steps (in square brackets is given the section where each step is described in detail):

1. Exclude spectral regions below 0 *ppm* and above 10 *ppm*, as well as the remaining water resonances from the original epilepsy spectra, obtaining a data set of 97 samples and 244 variables, the epilepsy data [8.3.1].
2. If the case is not MS244, row-scale the data to a constant total in the merged data set. This step was used in the conclusions of this chapter for comparative purposes with the data set without row-scaling, which was used in the experiments [8.3.4].

3. Log-transform (natural log) the epilepsy data [8.3.1].
4. Convert to positive definite the covariance matrix of the variables in the epilepsy data from step 2 [8.3.2].
5. Select a sample size for the reference and test data sets (100, 500 or 1000) [8.1].
6. Generate the reference data set using a random multivariate normal distribution generator with mean as the mean vector of the log data and dispersion matrix as the covariance matrix obtained in step 3 [8.3.2].
7. Select the number of variables to mean-shift, from 244, 120, 20, 3 and 1 [8.1].
8. Select the method of choosing which variables to be mean-shifted using the MAXDEV, MINDEV or MAXMEAN method [8.1].
9. Select the offset for the mean-shifting of the chosen variables [8.3.3].
10. Obtain the shifted mean vector [8.3.3].
11. Generate the test data set similarly to step 5 but using as mean the shifted mean vector [8.3.3].
12. Merge the two artificially generated data sets to one [8.3.5].
13. Column-scale (weighted centring) the merged data set [8.3.5].
14. Perform PCA on the merged data set of step 13 [8.3.6].
15. Apply Linear Discriminant analysis (LDA) to the two artificial data sets using the scores of the first two PCs, and calculate the *LDA misclassification rates* and the *average separation* between the two data sets from the PCA scores [8.3.7].
16. Repeat steps 4 – 15 a pre-selected number of times (100).
17. Calculate average values of the misclassification rates of LDA in 100 runs and plot these rates versus offsets superimposed with the vertical error bars of the misclassification rates at each offset point in the plot [8.3.7].
18. Calculate average values of the average separation in 100 runs and plot the average separation versus offsets superimposed with the vertical error bars of the average separation at each offset point in the plot [8.3.6.1].
19. Compute the coefficient of variation for the two statistics, corresponding to the error bars at each offset point in the plot [8.3.6.1].

Function `runSimulation()`, on page 322, is the main function of the simulation algorithm running all previously mentioned R functions to perform the 19 steps described above. In addition, function `plotBoundaries()` [page 326] plots a LDA boundary for the two sets in each of the simulation experiments, such as Figure 8.10 and function `plotMeanShifting()` [on page 324] creates a plot of the comparison of the mean spectra of the two data sets in each simulation experiment, such as Figure 8.9.

## 8.4 Simulation Experiments and Results

### 8.4.1 Introduction

In this section the results of a number of simulation experiments are described in detail. The experiments are divided into four subsections. The first is the special case MS244, as it is the same for all the variable selection methods, and the other three cover the experiments using the methods MAXDEV, MINDEV and MAXMEAN in this order. The results in MAXDEV are given in detail in the second subsection with the cases MS120, MS20, MS3 and MS1 in this order. The last two subsections state briefly the main findings and the results are given in the appendices. The results of each experiment include the following in this order:

- An illustration of the mean-shifting procedure, by plotting the mean spectra of the two data sets in S500 and an offset chosen to correspond to a misclassification rate of  $\approx 0\%$ .
- A table with the *misclassification rates* and the *Average separation* for the three sample size cases (S100, S500, S1000) using offsets such that the misclassification rate is approximately in the range 0 – 30 % in 100 runs of an experiment.
- Principal components scores plots, for offsets corresponding to misclassification rates of 20 %, 10 % and 1 % in sample size case S500, superimposed with the LDA boundary of the two data sets.
- Plots of the two statistics (*misclassification rate* and *average separation*) versus offsets for the three sample size cases, superimposed with vertical error bars, in 100 runs of an experiment.
- A table containing the coefficient of variation values for the two statistics to compute and compare the dispersion of the two stats in each offset value and sample size case.
- At the end of the MS244 and of the MAXDEV sections, a summary table for the offsets required in all subsets of variables (MS120, MS20, MS3 and MS1) and sample size cases (S100, S500 and S1000) to achieve *misclassification rates* of 20, 15, 10, 5, and 1 % in 100 runs of an experiment.

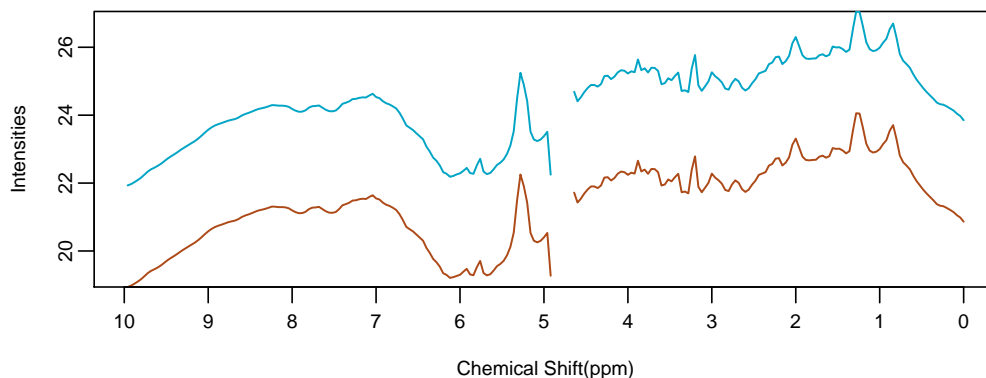
All the offsets in this section are given on the original scale of the data (multiplicative offsets) and the values on the y axis of the spectra plots are log-transformed (but not row-scaled) intensities.

### 8.4.2 Case MS244

There is no variable selection in case MS244, since all variables in the data set are mean-shifted. An example of the mean-shifting procedure for case MS244, with S500



and offset 20.09 (corresponding to a misclassification rate of  $\approx 0\%$ ), can be seen in Figure 8.9. Experiments in all three sample size cases showed that the two data sets are



**Figure 8.9:** Illustration of the mean-shifting procedure in the case MS244 with S500 and offset 20.09. The brown and blue lines are the mean spectra of the reference and test data set respectively. The mean-shifted variables correspond to the points in the spectra for which the blue line is above the brown.

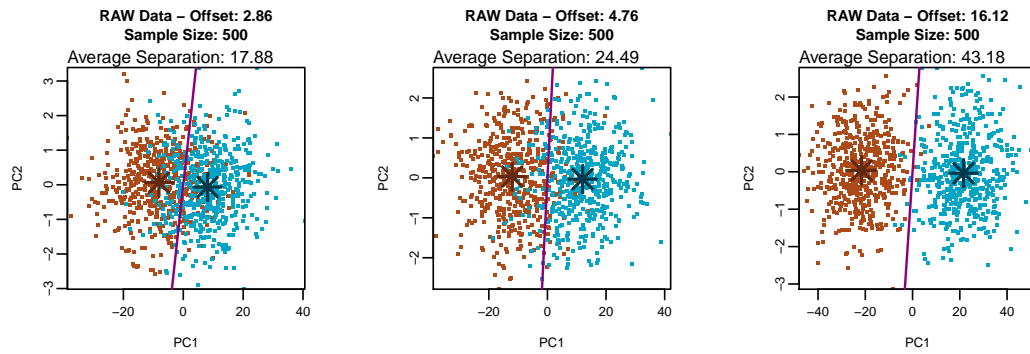
linearly separated for offsets above 20.09, with LDA misclassification rates below 1% and average separation value of  $\approx 47$ . Table 8.2 gives the misclassification rates and average separation values of the experiments in the cases S100, S500 and S1000 respectively, for offset values in the range 2.23 – 20.09. From this table it can be seen that in all sample size cases offsets in the range 2.23 – 20.09 are required to achieve misclassification rates of  $\approx 25 - 0.5\%$  respectively. Similarly, the average separation between the two data sets is  $\approx 15$  for a 20% misclassification rate and the two data sets are almost linearly separable when the average separation is  $\approx 17$  with misclassification rate less than 1%, as expected. In general, there are no great differences among the three sample size cases with respect to the *misclassification rate*, *average separation* and the offsets required to obtain those statistics (e.g. for offset 2.23, the misclassification rates are 25.63%, 25.60% and 25.71%, the average separation values are 15.35, 15.37 and 15.41 in cases S100, S500 and S1000 respectively, and for offset 20.09 the misclassification rates are 0.52%, 0.81% and 0.78% and the average separation values are 47.06, 46.82 and 46.90 for the same simulation cases).

An illustration of how the mean-shifting procedure affects the capability of PCA to discriminate the two data sets can be seen in Figure 8.10, for offsets 2.86, 4.76 and 16.12, which correspond to 20%, 10% and 1% misclassification rates respectively. A graphical representation of the relation between offsets and the two statistics can be seen in Figure 8.11. These plots confirm the findings from Table 8.2 concerning the sample size of the two data sets. That is, the sample size does not play any role in the offsets, as the required offsets to achieve misclassification rates of 0 – 20% are similar in all three sample size cases.

**Table 8.2:** Average LDA misclassification rates and average separation values for the case MS244 in 100 runs of the experiment.

<b>S100</b>						
<b>Offset</b>	2.23	2.72	3.32	4.06	4.95	6.05
<b>Error Rate (%)</b>	25.63	20.79	16.90	12.50	9.57	7.18
<b>Average Separation</b>	15.35	17.39	19.78	22.66	25.56	28.37
<b>Offset</b>	7.39	9.03	11.02	13.46	16.44	20.09
<b>Error Rate (%)</b>	4.97	3.47	2.39	1.46	1.00	0.52
<b>Average Separation</b>	31.23	34.37	37.43	40.75	43.97	47.06
<b>S500</b>						
<b>Offset</b>	2.23	2.72	3.32	4.06	4.95	6.05
<b>Error Rate (%)</b>	25.60	21.09	16.62	12.85	9.78	7.23
<b>Average Separation</b>	15.37	17.49	19.95	22.67	25.52	28.42
<b>Offset</b>	7.39	9.03	11.02	13.46	16.44	20.09
<b>Error Rate (%)</b>	5.25	3.75	2.54	1.75	1.19	0.81
<b>Average Separation</b>	31.39	34.55	37.61	40.68	43.72	46.82
<b>S1000</b>						
<b>Offset</b>	2.23	2.72	3.32	4.06	4.95	6.05
<b>Error Rate (%)</b>	25.71	21.00	16.78	12.95	9.85	7.34
<b>Average Separation</b>	15.41	17.47	19.95	22.64	25.41	28.36
<b>Offset</b>	7.39	9.03	11.02	13.46	16.44	20.09
<b>Error Rate (%)</b>	5.32	3.79	2.60	1.82	1.19	0.78
<b>Average Separation</b>	31.39	34.44	37.52	40.61	43.73	46.90

In addition, Figure 8.11 shows that for both statistics, the width of the error bar variation obtained in 100 runs reduces considerably as the sample size increases, indicating that the stability of the *misclassification rate* and the *average separation* depends on the sample size. The two statistics monotonically (but non-linearly) increase (in the case of the *average separation*) or decrease (in the case of the *misclassification rate*), as the offsets increase. To compare the dispersion of the two statistics in each offset over 10 runs, the coefficient of variation will be computed. Table 8.3 gives the results of the computations for the CV, the standard deviation and the mean of the two statistics for each of the selected offsets. It is clear that the dispersion of the *average separation* is far smaller than that of the *misclassification rate*, decreasing as the offsets decrease, whereas in the case of *misclassification rate*, as the mean of the error decreases towards 1, the CV increases considerably, especially for large offset values. As is logical, the larger the offset, the larger the distance is between the two data sets; that is, the larger the mean of the *average separation* is, and the CV decreases as the offsets increase. The CV values for both statistics are definitely affected by the increase in the sample size, with both statistics showing far smaller dispersion in the S500 and S1000 cases. In general, in the case MS244, the *average separation* is far more stable than the *misclassification*



**Figure 8.10:** Visualisation of the LDA boundaries using the first two PCs of the two artificial data sets in the case MS244. The reference and test data points are depicted in brown and blue respectively.

rate, as the CV values for both statistics indicate.

A summary of the offsets required to achieve misclassification rates of 20, 15, 10, 5 and 1 % for all sample size cases can be seen in Table 8.4. It can be concluded that for PCA to discriminate between the two data sets giving a misclassification rate of 1 % or less, an approximately 16-fold increase of all 244 variables is required. This size of increase is not practically feasible. Therefore, PCA cannot discriminate between the two data sets in the case MS244 given the means and variances used in the simulation, regardless of the data sets' sample size.

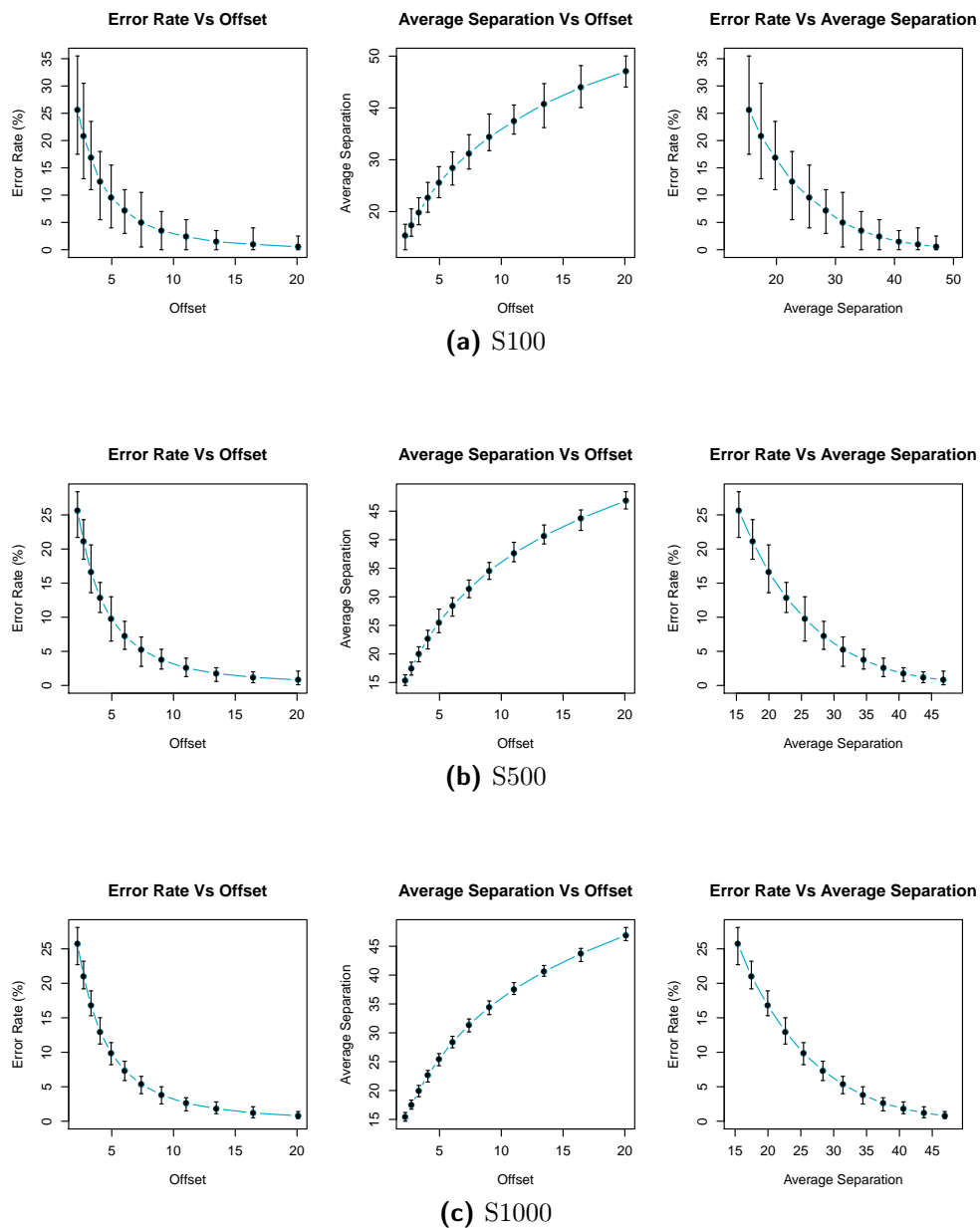
### 8.4.3 Maximum Deviation (MAXDEV)

#### Introduction

The MAXDEV method chooses a specific subset of variables to mean-shift in decreasing order of size of their standard deviation. This will be applied to simulation experiments with subsets of 120, 20, 3 and 1 variables for all three sample size cases, S100, S500 and S1000.

#### Case MS120

The mean-shifting procedure for case MS120, with S500 and offset 1.55 (corresponding to a misclassification rate of  $\approx 0\%$ ), can be seen in Figure 8.12. The 120 mean-shifted variables and their standard deviation can be seen in Table A.1 in decreasing order of standard deviation. Experiments in all three sample size cases showed that the two data sets are linearly separated for offsets above 1.55, with LDA misclassification rates below 1% and average separation value of  $\approx 12.2$ . Table 8.5 gives the misclassification rates and average separation values of the experiments in the cases S100, S500 and S1000 respectively, for offsets in the range 1.25 – 1.55. It can be seen that offsets in the range 1.25 – 1.55 are required in all sample size cases, to achieve misclassification rates



**Figure 8.11:** Graphical representation of the relation among *LDA misclassification rates*, *average separation* and *offsets* in the case MS244. The blue lines represent the mean values of each statistic for each offset in the selected offset range. The offsets are the multiplicative factors on the original scale of the data. The vertical error bars are such that the top and bottom of a bar correspond to the maximum and minimum statistic value at the respective offset. The statistics values are the average values in 100 runs of the experiment. The two average separation plots in each sample size case are drawn using the same range of values for the *average separation*.

**Table 8.3:** Coefficient of variation results for case MS244, of the *LDA misclassification rates* and *average separation* values in 100 runs of the experiment.

S100						
Offset	2.23	2.72	3.32	4.06	4.95	6.05
Error Rate (CV)	13.06	14.73	14.61	17.77	21.14	25.01
Error Rate (StDev)	3.34	3.06	2.47	2.22	2.02	1.79
Error Rate (Mean)	25.63	20.79	16.90	12.50	9.57	7.18
Average Separation (CV)	6.82	6.26	5.88	5.34	4.97	5.09
Average Separation (StDev)	1.04	1.08	1.16	1.21	1.27	1.44
Average Separation (Mean)	15.35	17.39	19.78	22.66	25.56	28.37
Offset	7.39	9.03	11.02	13.46	16.44	20.09
Error Rate (CV)	32.28	43.67	41.46	73.60	90.17	125.80
Error Rate (StDev)	1.60	1.51	0.99	1.07	0.90	0.66
Error Rate (Mean)	4.97	3.47	2.39	1.46	1.00	0.52
Average Separation (CV)	4.21	4.41	3.42	3.74	3.46	2.79
Average Separation (StDev)	1.31	1.51	1.28	1.52	1.52	1.31
Average Separation (Mean)	31.23	34.37	37.43	40.75	43.97	47.06
S500						
Offset	2.23	2.72	3.32	4.06	4.95	6.05
Error Rate (CV)	4.86	5.33	7.28	7.65	11.54	12.36
Error Rate (StDev)	1.24	1.13	1.21	0.98	1.13	0.89
Error Rate (Mean)	25.60	21.10	16.63	12.86	9.79	7.23
Average Separation (CV)	2.73	2.55	2.65	2.48	2.56	2.01
Average Separation (StDev)	0.42	0.45	0.53	0.56	0.65	0.57
Average Separation (Mean)	15.37	17.49	19.96	22.68	25.53	28.43
Offset	7.39	9.03	11.02	13.46	16.44	20.09
Error Rate (CV)	13.52	17.94	19.35	20.39	26.81	40.65
Error Rate (StDev)	0.71	0.67	0.49	0.36	0.32	0.33
Error Rate (Mean)	5.26	3.76	2.55	1.76	1.20	0.81
Average Separation (CV)	1.93	1.92	1.54	1.59	1.46	1.35
Average Separation (StDev)	0.61	0.66	0.58	0.65	0.64	0.63
Average Separation (Mean)	31.39	34.56	37.61	40.68	43.73	46.82
S1000						
Offset	2.23	2.72	3.32	4.06	4.95	6.05
Error Rate (CV)	3.95	3.95	4.40	5.84	7.24	7.99
Error Rate (StDev)	1.01	0.83	0.74	0.76	0.71	0.59
Error Rate (Mean)	25.71	21.00	16.79	12.95	9.85	7.34
Average Separation (CV)	1.85	1.88	1.92	1.69	1.61	1.39
Average Separation (StDev)	0.29	0.33	0.38	0.38	0.41	0.39
Average Separation (Mean)	15.42	17.47	19.96	22.65	25.41	28.36
Offset	7.39	9.03	11.02	13.46	16.44	20.09
Error Rate (CV)	9.56	11.56	13.51	14.97	23.88	24.34
Error Rate (StDev)	0.51	0.44	0.35	0.27	0.29	0.19
Error Rate (Mean)	5.33	3.80	2.60	1.82	1.19	0.79
Average Separation (CV)	1.46	1.27	1.05	0.95	1.06	0.95
Average Separation (StDev)	0.46	0.44	0.39	0.38	0.46	0.45
Average Separation (Mean)	31.40	34.45	37.52	40.62	43.73	46.91

of  $\approx 25 - 0.1\%$  respectively. Similarly, the average separation between the two data sets is  $\approx 11.3$  for a  $20\%$  misclassification rate and the two data sets are almost linearly separable when the average separation is  $\approx 12.2$ , with misclassification rate less than  $1\%$  as expected. In general, there are no noticeable differences among the three sample

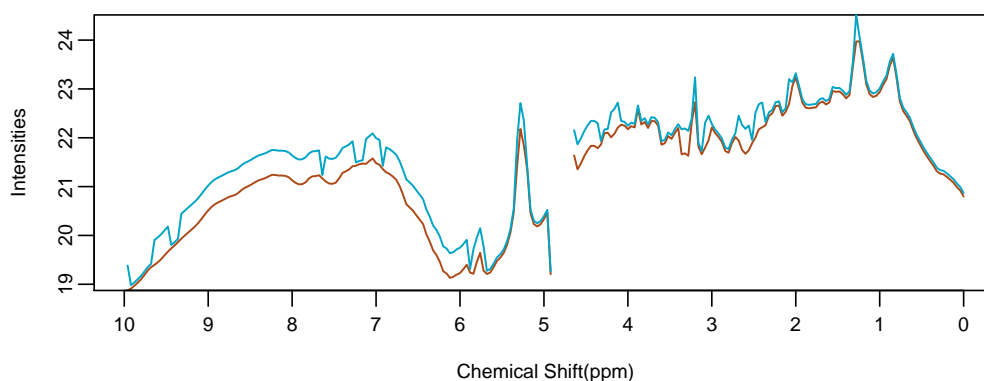
**Table 8.4:** Summary results (offsets) for the *LDA misclassification rates* in the case MS244. The results are for 100 runs of the simulation algorithm and the offsets correspond to multiplicative factors on the original scale of the data. An offset is the value above which a selected misclassification rate percentage is achieved, e.g in case S500, at most 10% of the samples are misclassified when the offset is 4.75 or above.

Subset of Variables	Sample Size	Misclassification Rate				
		20%	15%	10%	5%	1%
MS244	S100	2.85	3.49	4.71	7.61	15.64
	S500	2.85	3.56	4.75	7.69	16.11
	S1000	2.85	3.56	4.80	7.69	16.60

size cases with respect to the *misclassification rate*, *average separation* and the offsets required to obtain those statistics.

An illustration of how the mean-shifting procedure affects the capability of PCA to discriminate the two data sets in this case can be seen in Figure 8.13, for offsets 1.27, 1.32 and 1.45 which correspond to 20 %, 10 % and 1 % misclassification rates respectively.

A graphical representation of the relation between offsets and the two statistics can be seen in Figure 8.14. Similarly to the case MS244, the sample size does not play any role in the offsets in case MS120, as the required offsets to achieve misclassification rates of 0 – 20 % are the same in all three sample size cases. The range of values in the two statistics obtained in 100 runs reduces significantly as the sample size increases, indicating that the stability of the *misclassification rate* values depends on the sample size. To assess whether this is true, Table 8.6 contains the results for the coefficient of variation in this case. As in case MS244, the dispersion of the *average separation*

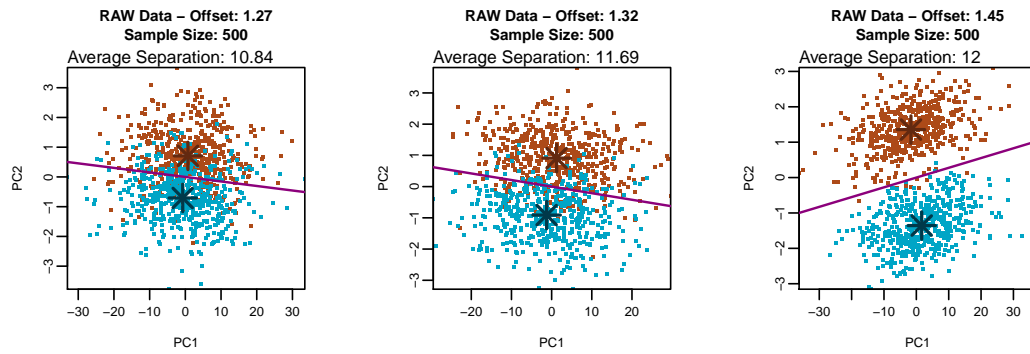


**Figure 8.12:** Illustration of the mean-shifting procedure with method MAXDEV in the case MS120 with S500 and offset 1.55. The brown and blue lines are the mean spectra of the reference and test data set respectively. The mean-shifted variables correspond to the points in the spectra for which the blue line is above the brown.

**Table 8.5:** Average LDA misclassification rates and average separation values for the case MS120, applying the MAXDEV method in 100 runs of the experiment.

<b>S100</b>						
<b>Offset</b>	1.25	1.27	1.30	1.32	1.35	1.38
<b>Error Rate (%)</b>	25.10	20.29	15.05	11.01	7.54	4.57
<b>Average Separation</b>	11.28	11.38	11.40	11.54	11.63	11.65
<b>Offset</b>	1.40	1.43	1.46	1.49	1.52	1.55
<b>Error Rate (%)</b>	2.50	1.68	0.87	0.31	0.18	0.09
<b>Average Separation</b>	11.70	11.79	11.80	11.96	12.18	12.23
<b>S500</b>						
<b>Offset</b>	1.25	1.27	1.30	1.32	1.35	1.38
<b>Error Rate (%)</b>	25.24	20.05	15.49	10.96	7.19	4.87
<b>Average Separation</b>	11.37	11.32	11.44	11.49	11.65	11.70
<b>Offset</b>	1.40	1.43	1.46	1.49	1.52	1.55
<b>Error Rate (%)</b>	2.89	1.59	0.87	0.53	0.22	0.11
<b>Average Separation</b>	11.79	11.83	11.94	12.00	12.09	12.15
<b>S1000</b>						
<b>Offset</b>	1.25	1.27	1.30	1.32	1.35	1.38
<b>Error Rate (%)</b>	25.27	20.02	15.60	11.10	7.41	4.73
<b>Average Separation</b>	11.31	11.34	11.42	11.53	11.57	11.63
<b>Offset</b>	1.40	1.43	1.46	1.49	1.52	1.55
<b>Error Rate (%)</b>	2.84	1.63	0.92	0.48	0.23	0.12
<b>Average Separation</b>	11.73	11.81	11.91	12.01	12.06	12.18

is far smaller than that of the *misclassification rate*, decreasing as the offsets decrease, whereas in the case of *misclassification rate* the CV increases considerably, especially for large offset values. An interesting observation is that comparing the CV values for the two statistics in the cases MS244 and MS120 using MAXDEV, despite the reduction of mean-shifted variables from 244 to 120, the CV of the *misclassification rate* is considerably larger in case MS120 for all three sample size cases and all offsets, whereas the CV of the *average separation* is slightly smaller in case MS120 for small offsets and slightly larger for large offsets than in case MS244. In general, the sample size clearly affects the CV values of both statistics independently of the number of mean-shifted variables, as the larger the sample size is, the smaller the CV values of both statistics for all offsets. In addition, the standard deviation and mean of the *average separation* error bars in Figure 8.14, are very consistent, with their values for all offsets being very close to each other in the three sample size cases (especially in the cases S500 and S1000), whereas in the case of *misclassification rate* the standard deviation reduces considerably as the sample size and the offsets increase, and its mean, as expected, decreases monotonically. Thus, as in the case MS244, in the case MS120, the *average separation* error bar values are far more consistent, stable and far less affected than the



**Figure 8.13:** Visualisation of the LDA boundaries for the two artificial data sets in the case MS120 (MAXDEV)). The data corresponds to the first two PCs for LDA. The reference and test data points are depicted in brown and blue respectively.

*misclassification rate.*

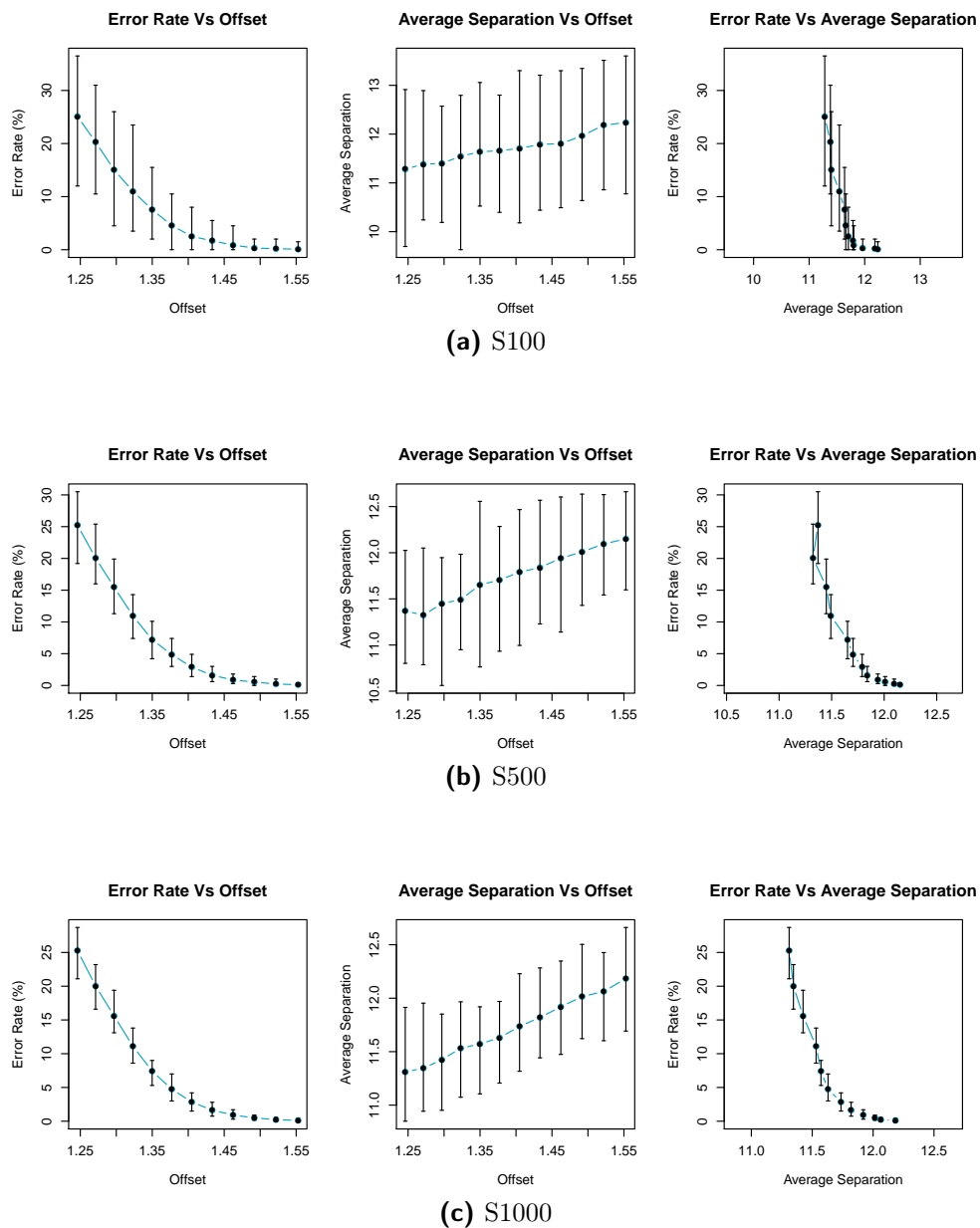
For PCA to discriminate between the two data sets with a misclassification rate of 1% or less, an approximate 50% increase (the multiplicative offset is 1.46) is required in the means of the selected 120 variables. Hence, it is possible for PCA to discriminate between the two data sets in case MS120 with MAXDEV, independently of the data sets' sample size. Thus, reducing the number of mean-shifted variables from 244 to 120, using the MAXDEV method, results in smaller (feasible) offsets and average separation values required in the simulation experiments.

### Case MS20

The mean-shifting procedure for case MS20, with S500 and offset 2.18 (corresponding to a misclassification rate of  $\approx 0\%$ ), can be seen in Figure 8.15. The 20 mean-shifted variables and their standard deviation can be seen in Table A.1 in decreasing order of standard deviation. Experiments using offsets in the range 1.40–2.18 in all three sample size cases showed that the two data sets are linearly separated for offsets above 2.05, with LDA misclassification rates below 1% and average separation value of  $\approx 11.8$ . Table 8.7 shows the misclassification rates and average separation values of the experiments in the cases S100, S500 and S1000 respectively, for offsets in the range 1.40 – 2.18. From the table it can be seen that offsets in the range 1.52 – 2.05 are required in all sample size cases, to achieve misclassification rates of  $\approx 20 - 0.1\%$  respectively. Similarly, the average separation between the two data sets is  $\approx 11.1$  for a 20% misclassification rate and the two data sets are almost linearly separable when the average separation is  $\approx 11.8$ , with misclassification rate less than 1% as expected. In general, there are no big differences among the three sample size cases with respect to the *misclassification rate*, *average separation* and offsets required to obtain those statistics.

An illustration of how the mean-shifting procedure affects the capability of PCA to discriminate the two data sets in this case can be seen in Figure 8.16, for offsets 1.52, 1.62





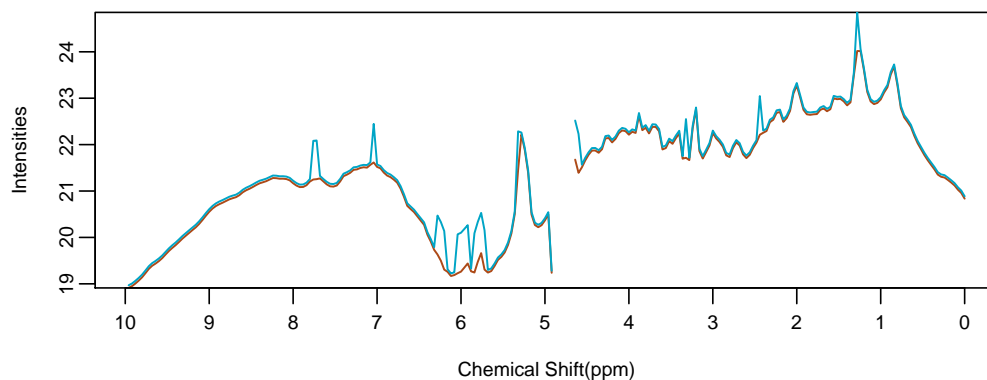
**Figure 8.14:** Graphical representation of the relation among *LDA misclassification rates*, *average separation* and *offset* in the case MS120 for method MAXDEV. The blue lines represent the mean values of each statistic for each offset in the selected offset range. The offsets are the multiplicative factors on the original scale of the data. The vertical error bars are such that the top and bottom of a bar correspond to the maximum and minimum statistic value at the respective offset. The statistics values are the average values in 100 runs of the experiment. The two average separation plots in each sample size case, are drawn using the same range of values for the *average separation*.

**Table 8.6:** Coefficient of variation results for case MS120 using method MAXDEV, of the *LDA* misclassification rates and average separation values in 100 runs of the experiment.

S100						
Offset	1.25	1.27	1.30	1.32	1.35	1.38
Error Rate (CV)	19.93	20.33	28.68	32.12	36.38	47.58
Error Rate (StDev)	5.00	4.13	4.32	3.54	2.75	2.17
Error Rate (Mean)	25.10	20.30	15.05	11.01	7.54	4.57
Average Separation (CV)	5.48	5.15	4.70	5.18	4.94	4.58
Average Separation (StDev)	0.62	0.59	0.54	0.60	0.58	0.53
Average Separation (Mean)	11.28	11.38	11.40	11.54	11.64	11.66
Offset	1.40	1.43	1.46	1.49	1.52	1.55
Error Rate (CV)	64.29	78.97	108.87	165.26	219.52	288.84
Error Rate (StDev)	1.61	1.33	0.95	0.52	0.41	0.26
Error Rate (Mean)	2.50	1.68	0.87	0.32	0.18	0.09
Average Separation (CV)	5.69	4.90	4.74	4.91	4.35	4.88
Average Separation (StDev)	0.67	0.58	0.56	0.59	0.53	0.60
Average Separation (Mean)	11.71	11.79	11.80	11.96	12.18	12.24
S500						
Offset	1.25	1.27	1.30	1.32	1.35	1.38
Error Rate (CV)	8.29	9.98	11.10	13.73	17.93	18.35
Error Rate (StDev)	2.09	2.00	1.72	1.51	1.29	0.89
Error Rate (Mean)	25.25	20.06	15.49	10.96	7.20	4.87
Average Separation (CV)	1.96	2.06	2.32	2.15	2.56	2.24
Average Separation (StDev)	0.22	0.23	0.27	0.25	0.30	0.26
Average Separation (Mean)	11.37	11.32	11.45	11.49	11.65	11.70
Offset	1.40	1.43	1.46	1.49	1.52	1.55
Error Rate (CV)	26.46	32.28	36.44	52.48	81.06	104.49
Error Rate (StDev)	0.77	0.52	0.32	0.28	0.18	0.12
Error Rate (Mean)	2.90	1.60	0.88	0.53	0.23	0.12
Average Separation (CV)	2.35	2.19	2.14	1.98	1.97	2.02
Average Separation (StDev)	0.28	0.26	0.26	0.24	0.24	0.25
Average Separation (Mean)	11.79	11.84	11.94	12.01	12.10	12.15
S1000						
Offset	1.25	1.27	1.30	1.32	1.35	1.38
Error Rate (CV)	6.00	6.94	8.05	9.40	9.96	16.63
Error Rate (StDev)	1.52	1.39	1.26	1.04	0.74	0.79
Error Rate (Mean)	25.28	20.02	15.60	11.10	7.42	4.73
Average Separation (CV)	1.71	1.77	1.51	1.61	1.47	1.37
Average Separation (StDev)	0.19	0.20	0.17	0.19	0.17	0.16
Average Separation (Mean)	11.31	11.35	11.42	11.53	11.57	11.63
Offset	1.40	1.43	1.46	1.49	1.52	1.55
Error Rate (CV)	17.70	25.11	27.29	32.85	52.15	67.27
Error Rate (StDev)	0.50	0.41	0.25	0.16	0.12	0.08
Error Rate (Mean)	2.85	1.63	0.93	0.49	0.23	0.12
Average Separation (CV)	1.59	1.45	1.42	1.30	1.47	1.48
Average Separation (StDev)	0.19	0.17	0.17	0.16	0.18	0.18
Average Separation (Mean)	11.74	11.82	11.92	12.01	12.06	12.18

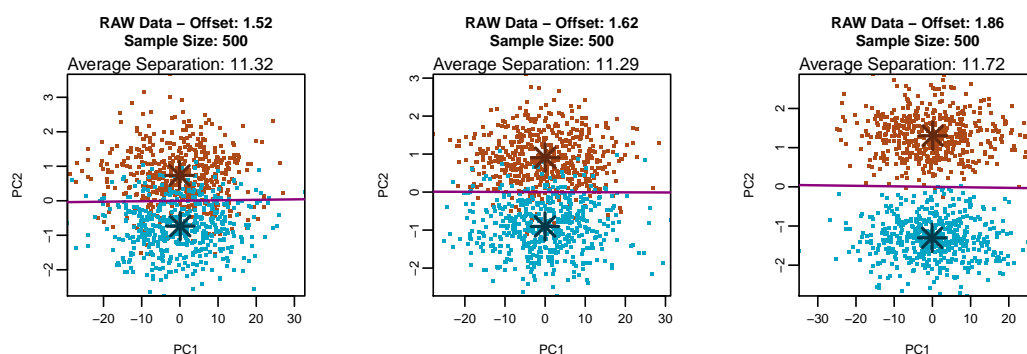
and 1.86, which correspond to 20%, 10% and 1% misclassification rates respectively.

The relation among *misclassification rates*, *average separation* and offsets in the case MS20 in 100 runs is shown in Figure 8.17. It can be seen that the change in the average separation with offsets is much less than the change in the misclassification rate. Similarly to cases MS244 and MS120, the sample size does not play any role

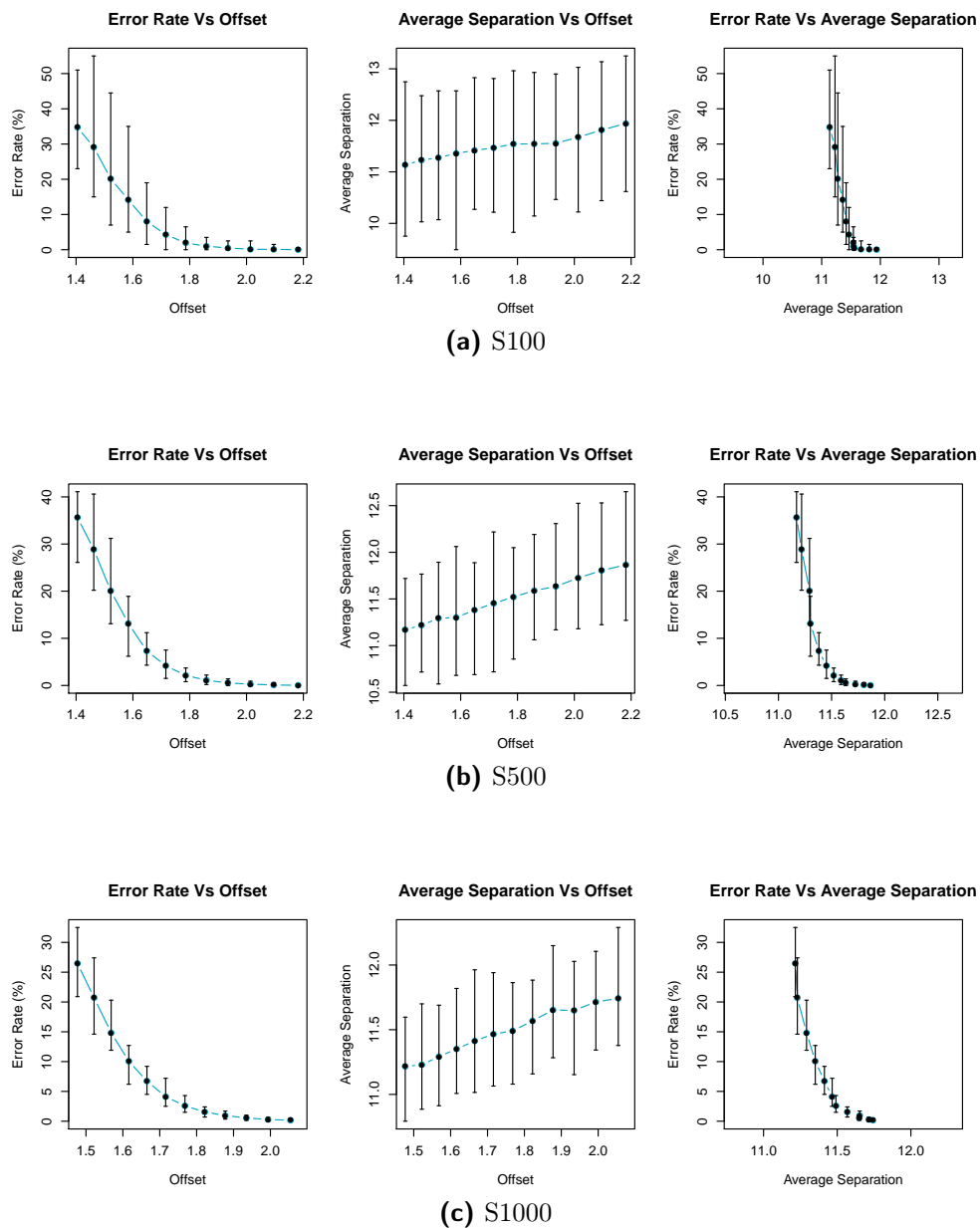


**Figure 8.15:** Illustration of the mean-shifting procedure with method MAXDEV in the case MS20 with S500 and offset 2.18. The brown and blue lines are the mean spectra of the reference and test data set respectively. The mean-shifted variables correspond to the points in the spectra for which the blue line is above the brown.

in the offsets in case MS20, as the required offsets to achieve misclassification rates of 0 – 20% are similar in all three sample size cases. The range of values in the two statistics obtained in 100 runs again reduces noticeably as the sample size increases, indicating that the stability of the *misclassification rate* depend on the sample size. Table 8.8 contains the results for the coefficient of variation, the standard deviation and the mean values of the error bars in Figure 8.17, for the two statistics in the case MS20. The table shows clearly that the values of CV, standard deviation and mean of the error bars for the *average separation* do not differ in general from those of case MS120, being also very stable between all offsets (especially the mean, which in all three samples size cases has values in the range 11.1 - 12). On the other hand, the values of the same statistics for the *misclassification rate* indicate that this criterion is affected by the number of mean-shifted variables, as the values of CV, standard deviation and



**Figure 8.16:** Visualisation of the LDA boundaries for the two artificial data sets in the case MS20 (MAXDEV). The data corresponds to the first two PCs for LDA. The reference and test data points are depicted in brown and blue respectively.



**Figure 8.17:** Graphical representation of the relation among *LDA misclassification rates*, *average separation* and *offset* in the case MS20 applying the MAXDEV method. The blue lines represent the mean values of each statistic for each offset in the selected offset range. The offsets are the multiplicative factors on the original scale of the data. The vertical error bars are such that the top and bottom of a bar correspond to the maximum and minimum statistic value at the respective offset. The statistics values are the average values in 100 runs of the experiment. The two average separation plots in each sample size case are drawn using the same range of values for the *average separation*.

**Table 8.7:** Average LDA misclassification rates and average separation values for the case MS20, applying the MAXDEV method in 100 runs of the experiment.

<b>S100</b>						
<b>Offset</b>	1.40	1.46	1.52	1.58	1.65	1.72
<b>Error Rate (%)</b>	34.74	29.08	20.14	14.13	8.04	4.27
<b>Average Separation</b>	11.13	11.22	11.27	11.35	11.41	11.46
<b>Offset</b>	1.79	1.86	1.93	2.01	2.10	2.18
<b>Error Rate (%)</b>	1.94	0.99	0.49	0.12	0.09	0.01
<b>Average Separation</b>	11.54	11.54	11.55	11.67	11.81	11.93
<b>S500</b>						
<b>Offset</b>	1.40	1.46	1.52	1.58	1.65	1.72
<b>Error Rate (%)</b>	35.62	28.83	20.07	13.16	7.31	4.18
<b>Average Separation</b>	11.17	11.21	11.29	11.30	11.38	11.45
<b>Offset</b>	1.79	1.86	1.93	2.01	2.10	2.18
<b>Error Rate (%)</b>	2.12	1.12	0.52	0.28	0.09	0.03
<b>Average Separation</b>	11.52	11.58	11.63	11.72	11.80	11.86
<b>S1000</b>						
<b>Offset</b>	1.48	1.52	1.57	1.62	1.67	1.72
<b>Error Rate (%)</b>	26.47	20.70	14.83	10.05	6.72	4.10
<b>Average Separation</b>	11.21	11.22	11.29	11.35	11.41	11.46
<b>Offset</b>	1.77	1.82	1.88	1.93	1.99	2.05
<b>Error Rate (%)</b>	2.58	1.54	0.95	0.53	0.29	0.15
<b>Average Separation</b>	11.49	11.56	11.65	11.64	11.71	11.74

mean are in general higher than those of MS120 and MS244. This is also logical, as the method for selecting the 20 variables is MAXDEV. As in the cases MS244 and MS120, the dispersion of the *average separation* is far smaller than that of the *misclassification rate*, decreasing slightly but not monotonically, as the offsets decrease, whereas in the case of *misclassification rate* the CV increases considerably, especially for large offset values. An interesting observation is that comparing the CV values for the two statistics in the cases MS244, MS120 and MS20 using MAXDEV, despite the reduction of mean-shifted variables from 244 to 20, the CV of the *misclassification rate* is generally larger in case MS20 for all three sample size cases and all offsets, whereas the CV of the *average separation* is slightly smaller in case MS20 for small offsets and slightly larger for large offsets than in the other two cases. The sample size affects the CV values of both statistics independently of the number of mean-shifted variables, as the larger the sample size is, the smaller the CV values of both statistics in all offsets. However, the *misclassification rate* is affected much more than the *average separation* from the sample size and the number of mean-shifted variables. In addition, the standard deviation and mean of the *average separation* error bars in Figure 8.17, are very consistent, with their values for all offsets being very close to each other in the three sample size cases

**Table 8.8:** Coefficient of variation results for case MS20 using method MAXDEV, of the *LDA* misclassification rates and average separation values in 100 runs of the experiment.

S100						
Offset	1.40	1.46	1.52	1.58	1.65	1.72
Error Rate (CV)	18.80	24.95	36.26	41.33	40.43	54.63
Error Rate (StDev)	6.53	7.26	7.30	5.84	3.25	2.33
Error Rate (Mean)	34.74	29.09	20.14	14.13	8.04	4.27
Average Separation (CV)	5.37	5.15	4.71	5.21	4.92	4.58
Average Separation (StDev)	0.60	0.58	0.53	0.59	0.56	0.53
Average Separation (Mean)	11.14	11.23	11.27	11.36	11.42	11.47
Offset	1.79	1.86	1.93	2.01	2.10	2.18
Error Rate (CV)	76.78	108.91	128.81	302.67	286.77	703.53
Error Rate (StDev)	1.49	1.08	0.64	0.36	0.27	0.07
Error Rate (Mean)	1.94	1.00	0.50	0.12	0.10	0.01
Average Separation (CV)	5.90	4.63	4.32	4.75	4.07	4.51
Average Separation (StDev)	0.68	0.53	0.50	0.55	0.48	0.54
Average Separation (Mean)	11.54	11.54	11.55	11.67	11.81	11.94
S500						
Offset	1.40	1.46	1.52	1.58	1.65	1.72
Error Rate (CV)	8.42	13.21	15.90	18.03	19.90	26.52
Error Rate (StDev)	3.00	3.81	3.19	2.37	1.45	1.11
Error Rate (Mean)	35.63	28.83	20.07	13.16	7.31	4.18
Average Separation (CV)	2.13	2.08	2.00	1.98	2.24	2.17
Average Separation (StDev)	0.24	0.23	0.23	0.22	0.25	0.25
Average Separation (Mean)	11.17	11.22	11.29	11.30	11.38	11.45
Offset	1.79	1.86	1.93	2.01	2.10	2.18
Error Rate (CV)	30.12	41.07	55.08	59.16	130.71	169.91
Error Rate (StDev)	0.64	0.46	0.29	0.17	0.12	0.06
Error Rate (Mean)	2.12	1.12	0.52	0.28	0.09	0.04
Average Separation (CV)	2.09	2.11	2.02	2.22	2.37	2.12
Average Separation (StDev)	0.24	0.24	0.23	0.26	0.28	0.25
Average Separation (Mean)	11.52	11.59	11.63	11.72	11.80	11.87
S1000						
Offset	1.48	1.52	1.57	1.62	1.67	1.72
Error Rate (CV)	9.16	10.56	11.58	13.54	13.78	17.65
Error Rate (StDev)	2.43	2.19	1.72	1.36	0.93	0.73
Error Rate (Mean)	26.48	20.70	14.83	10.06	6.72	4.11
Average Separation (CV)	1.64	1.62	1.53	1.39	1.63	1.49
Average Separation (StDev)	0.18	0.18	0.17	0.16	0.19	0.17
Average Separation (Mean)	11.22	11.23	11.29	11.35	11.41	11.47
Offset	1.77	1.82	1.88	1.93	1.99	2.05
Error Rate (CV)	21.10	24.65	28.03	38.00	45.94	66.20
Error Rate (StDev)	0.55	0.38	0.27	0.20	0.13	0.10
Error Rate (Mean)	2.59	1.55	0.96	0.54	0.29	0.16
Average Separation (CV)	1.56	1.49	1.47	1.45	1.39	1.53
Average Separation (StDev)	0.18	0.17	0.17	0.17	0.16	0.18
Average Separation (Mean)	11.49	11.57	11.65	11.65	11.71	11.74

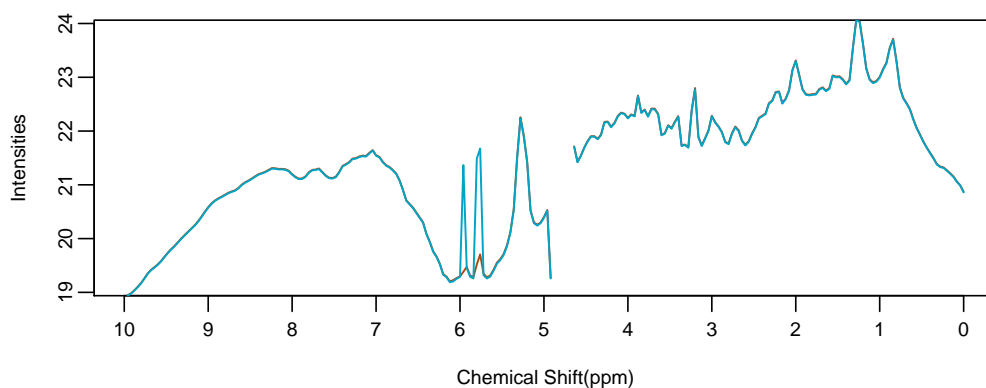
(especially in the cases S500 and S1000), whereas in the case of *misclassification rate* the standard deviation reduces considerably as the sample size and the offsets increase, and its mean, as expected, decreases monotonically. Thus, as in cases MS244 and MS120, also in case MS20, the *average separation* error bar values are far more consistent, stable and far less affected, than the *misclassification rate*.

For PCA to discriminate between the two data sets with a misclassification rate of

1% or less, an approximately 100% increase is required in the means of the 20 variables selected by MAXDEV. Reducing the number of mean-shifted variables from 244 to 20 using the MAXDEV method results in smaller (feasible) offsets and average separation values required.

### Case MS3

The mean-shifting procedure for case MS3, with S500 and offset 7.39 (corresponding to a misclassification rate of  $\approx 0\%$ ), can be seen in Figure 8.18. The 3 mean-shifted variables are 5.78, 5.82 and 5.98 and their standard deviations 0.876, 0.787 and 0.681 respectively (Table A.1). Simulation experiments using offsets in the range 2.46 – 7.77



**Figure 8.18:** Illustration of the mean-shifting procedure in the case MS3 with S500 and offset 7.39 (MAXDEV). The brown and blue lines are the mean spectra of the reference and test data set respectively. The mean-shifted variables correspond to the points in the spectra for which the blue line is above the brown. The 3 mean-shifted variables are at 5.78, 5.82 and 5.98 *ppm*.

in all three sample size cases showed that the two data sets are linearly separated for offsets above 7.39, with LDA misclassification rates below 1% and average separation value of  $\approx 11.9$ . Table 8.9 shows the results of the experiments in the cases S100, S500 and S1000 respectively, for offsets in the range 2.46 – 7.77. It can be seen that offsets of 2.46 – 7.77 are required in all sample size cases to achieve misclassification rates of  $\approx 25 - 0.5\%$  respectively. The average separation between the two data sets is  $\approx 11.3$  for a 20% misclassification rate and the two data sets are almost linearly separable when the average separation is  $\approx 11.9$ , with misclassification rate less than 1% as expected. In general, there are no great differences among the three sample size cases with respect to the *misclassification rate*, *average separation* and offsets required to obtain those statistics.

An illustration of the capability of PCA to discriminate the two data sets in this case can be seen in Figure 8.19, for offsets 2.83, 3.74 and 6.69 which correspond to 20%, 10% and 1% respectively. The relation among *misclassification rates*, *average*

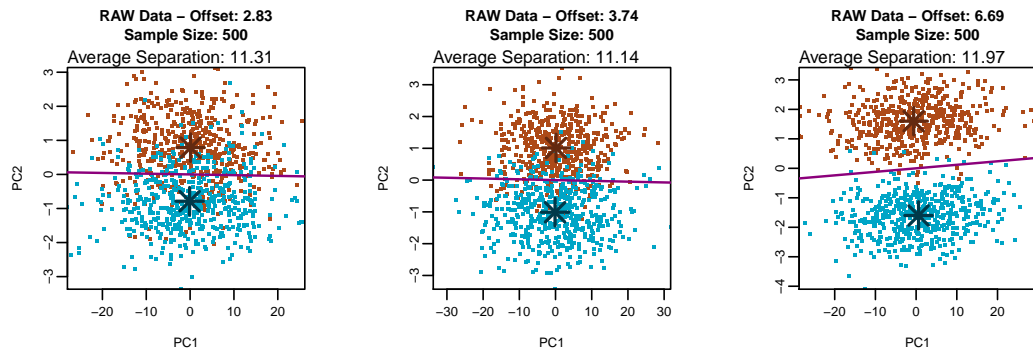
**Table 8.9:** Average LDA misclassification rates and average separation values for the case MS3, applying the MAXDEV method in 100 runs of the experiment.

<b>S100</b>						
<b>Offset</b>	2.46	2.72	3.00	3.32	3.67	4.06
<b>Error Rate (%)</b>	25.89	22.04	17.27	14.20	11.07	8.06
<b>Average Separation</b>	11.19	11.27	11.33	11.40	11.47	11.48
<b>Offset</b>	4.48	4.95	5.47	6.05	6.69	7.39
<b>Error Rate (%)</b>	5.47	4.25	2.80	1.65	1.09	0.52
<b>Average Separation</b>	11.54	11.58	11.59	11.71	11.85	11.94
<b>S500</b>						
<b>Offset</b>	2.46	2.72	3.00	3.32	3.67	4.06
<b>Error Rate (%)</b>	26.16	21.57	17.72	14.07	10.79	8.02
<b>Average Separation</b>	11.24	11.29	11.34	11.34	11.43	11.50
<b>Offset</b>	4.48	4.95	5.47	6.05	6.69	7.39
<b>Error Rate (%)</b>	5.81	3.95	2.92	1.94	1.23	0.78
<b>Average Separation</b>	11.55	11.63	11.66	11.76	11.84	11.90
<b>S1000</b>						
<b>Offset</b>	2.59	2.86	3.16	3.49	3.86	4.26
<b>Error Rate (%)</b>	23.85	19.79	15.95	12.42	9.32	6.86
<b>Average Separation</b>	11.23	11.29	11.37	11.43	11.47	11.50
<b>Offset</b>	4.71	5.21	5.75	6.36	7.03	7.77
<b>Error Rate (%)</b>	4.93	3.41	2.34	1.47	1.02	0.59
<b>Average Separation</b>	11.61	11.64	11.70	11.77	11.87	11.89

*separation* and offsets in the case MS3 in 100 runs, is visualised in Figure 8.20. As in the previous cases, the sample size does not play any role in the offsets in case MS3, as the required offsets to achieve misclassification rates of 0 – 20 % are similar in all three sample size cases. The range of values in the two statistics obtained in 100 runs reduces significantly as the sample size increases, indicating that the stability of the *misclassification rate* depends on the sample size. In the case MS3 with MAXDEV, average separation values are in the interval 9.5 – 13.1, for LDA misclassification rates between 0 and 40 %.

To examine whether the *average separation* values are more stable (less dispersed around its mean value) for each offset than the *misclassification rate* in all three sample size cases, the coefficient of variation will be computed for both statistics. Table 8.10 contains the results for the CV, standard deviation and mean values for the error bars corresponding to each of the two statistics for each offset, in all three sample size cases. As in all previous MS cases, the CV of the *misclassification rate* increases monotonically as the offsets increase, although for large sample sizes this increase is far less rapid, with the CV values being considerably smaller than in the case S100. The *average separation* is far less affected by the sample size, as although a decrease is observed

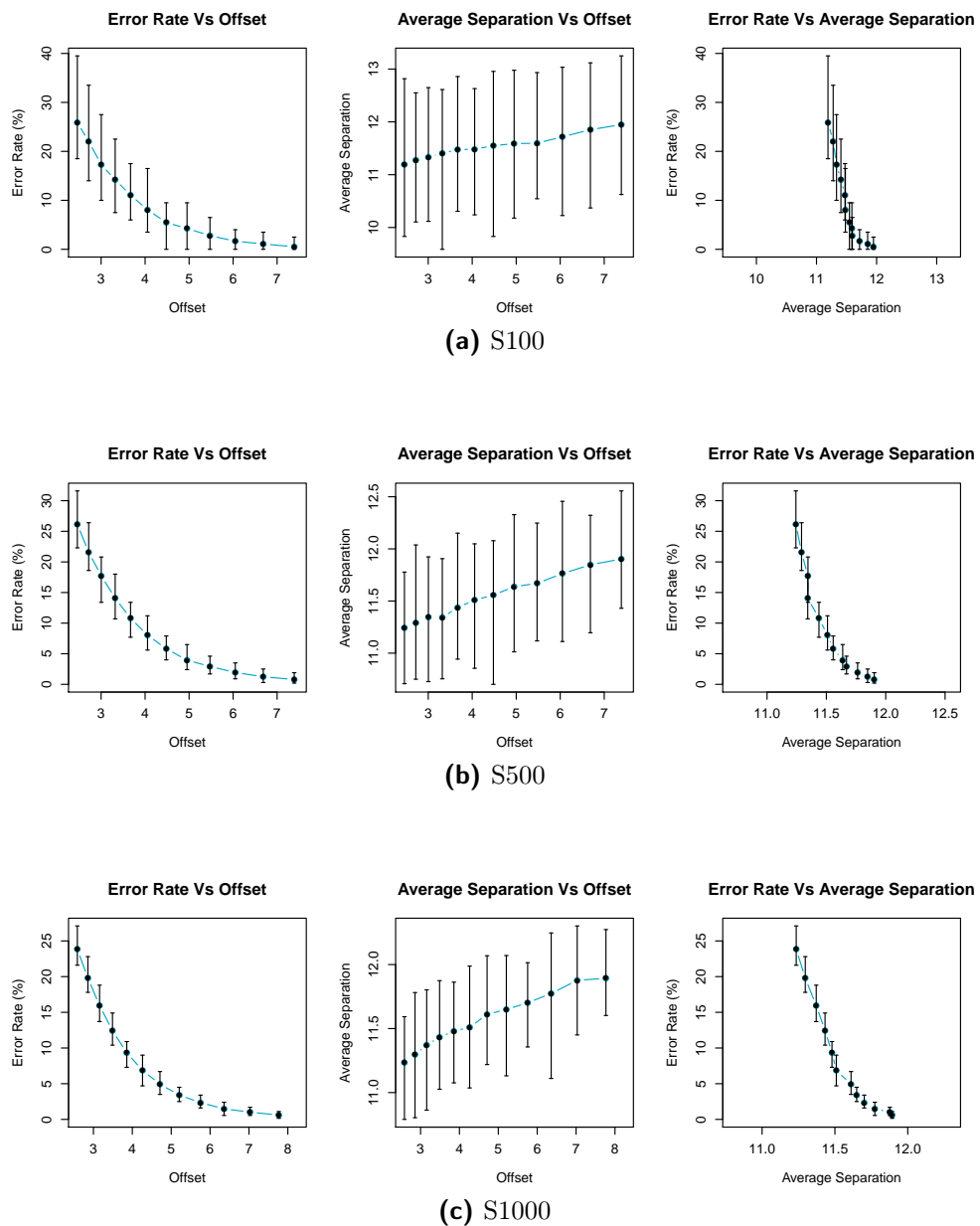




**Figure 8.19:** Visualisation of the LDA boundaries for the two artificial data sets in the case MS3 (MAXDEV). The data corresponds to the first two PCs for LDA. The reference and test data points are depicted in brown and blue respectively.

in its CV values for large sample sizes, this decrease is not nearly as large as in the case of the *misclassification rate*. However, the CV values of the *average separation* are inconsistent, since their increase or decrease do not depend much on the offsets in all three sample size cases. This is something that occurs in all MS cases with MAXDEV so far, whereas in the case MS244 the CV of *average separation* decreases far more consistently as the offsets increase. Therefore, it is clear that the average separation is affected considerably by the number of mean-shifted variables, with respect to the trend of the CV to decrease monotonically, and not the actual CV values, which are fairly consistent (very slightly decreasing) as the number of mean-shifted variables decreases. These facts are expected to be consistent also in the case MS1 using the MAXDEV method of selecting the mean-shifted variables. Comparing the CV results in Tables 8.3, 8.6, 8.8 and 8.10, it can be seen that the CV of the *misclassification rate* increases considerably in all sample size cases, as the number of variables decreases, up to but not including the MS3 case, where the CV values decrease again closely to those of the case MS244. Clearly, the number of mean-shifted variables affects the CV values of the error bars for the *misclassification rate*, with a number of mean-shifted variables using method MAXDEV, in the range 20 – 3 being the point at which the CV values start to decrease again towards those of MS244, independently of the sample size and the offsets. This is not true regarding the CV values of the *average separation*, as in this case the CV values in all selected subsets of mean-shifted variables (with the exception of the case MS244 where the CV values decreasing monotonically as the offsets increase), differ very slightly and in general, do not seem to be dependent on the offsets. It is clear that the number of mean-shifted variables affects mainly the *misclassification rate*, whereas the sample size affects both statistics in approximately equal measure.

The requirement for PCA to discriminate between the two data sets with a misclassification rate of 1% or less is an approximate seven-fold increase in the means of the 3 variables selected by MAXDEV (seen in Figure 8.18). This is clearly not acceptable as



**Figure 8.20:** Graphical representation of the relation among *LDA misclassification rates*, *average separation* and *offset* in the case MS3 applying the MAXDEV method. The blue lines represent the mean values of each statistic for each offset in the selected offset range. The offsets are the multiplicative factors on the original scale of the data. The vertical error bars are such that the top and bottom of a bar correspond to the maximum and minimum statistic value at the respective offset. The statistics values are the average values in 100 runs of the experiment. The two average separation plots in each sample size case are drawn using the same range of values for the *average separation*.

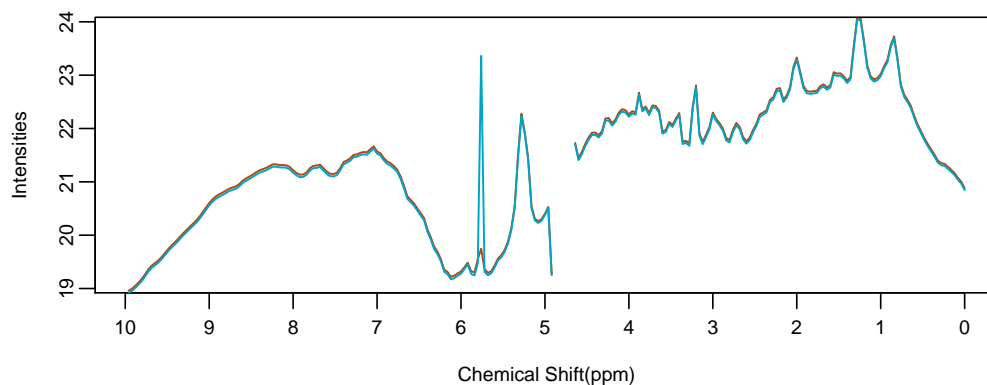
**Table 8.10:** Coefficient of variation results for case MS3 using method MAXDEV, of the *LDA* misclassification rates and average separation values in 100 runs of the experiment.

S100						
Offset	2.46	2.72	3.00	3.32	3.67	4.06
Error Rate (CV)	16.20	15.89	20.68	24.09	21.77	29.60
Error Rate (StDev)	4.19	3.50	3.57	3.42	2.41	2.39
Error Rate (Mean)	25.89	22.05	17.27	14.20	11.07	8.06
Average Separation (CV)	5.24	4.98	4.64	5.04	4.72	4.40
Average Separation (StDev)	0.59	0.56	0.53	0.58	0.54	0.51
Average Separation (Mean)	11.19	11.28	11.33	11.41	11.47	11.48
Offset	4.48	4.95	5.47	6.05	6.69	7.39
Error Rate (CV)	33.98	41.50	51.27	59.92	84.30	128.15
Error Rate (StDev)	1.86	1.77	1.44	0.99	0.92	0.67
Error Rate (Mean)	5.47	4.25	2.80	1.66	1.09	0.52
Average Separation (CV)	5.86	4.49	4.16	4.70	3.96	4.56
Average Separation (StDev)	0.68	0.52	0.48	0.55	0.47	0.55
Average Separation (Mean)	11.55	11.59	11.60	11.72	11.85	11.95
S500						
Offset	2.46	2.72	3.00	3.32	3.67	4.06
Error Rate (CV)	6.87	6.64	7.71	11.14	11.18	13.38
Error Rate (StDev)	1.80	1.43	1.37	1.57	1.21	1.07
Error Rate (Mean)	26.16	21.57	17.72	14.07	10.79	8.02
Average Separation (CV)	2.10	2.08	2.03	2.08	2.09	2.09
Average Separation (StDev)	0.24	0.23	0.23	0.24	0.24	0.24
Average Separation (Mean)	11.24	11.29	11.34	11.34	11.44	11.51
Offset	4.48	4.95	5.47	6.05	6.69	7.39
Error Rate (CV)	15.67	18.10	21.16	26.80	34.59	36.71
Error Rate (StDev)	0.91	0.72	0.62	0.52	0.43	0.29
Error Rate (Mean)	5.82	3.95	2.92	1.94	1.24	0.78
Average Separation (CV)	2.47	1.90	2.03	2.29	1.93	2.02
Average Separation (StDev)	0.29	0.22	0.24	0.27	0.23	0.24
Average Separation (Mean)	11.56	11.64	11.67	11.76	11.85	11.90
S1000						
Offset	2.59	2.86	3.16	3.49	3.86	4.26
Error Rate (CV)	5.18	5.52	7.21	7.14	8.15	10.20
Error Rate (StDev)	1.24	1.09	1.15	0.89	0.76	0.70
Error Rate (Mean)	23.85	19.79	15.96	12.43	9.32	6.86
Average Separation (CV)	1.38	1.45	1.55	1.51	1.56	1.76
Average Separation (StDev)	0.15	0.16	0.18	0.17	0.18	0.20
Average Separation (Mean)	11.23	11.30	11.37	11.43	11.48	11.51
Offset	4.71	5.21	5.75	6.36	7.03	7.77
Error Rate (CV)	11.36	13.29	15.57	20.64	24.45	30.01
Error Rate (StDev)	0.56	0.45	0.37	0.31	0.25	0.18
Error Rate (Mean)	4.93	3.41	2.35	1.48	1.02	0.60
Average Separation (CV)	1.50	1.50	1.26	1.62	1.39	1.18
Average Separation (StDev)	0.17	0.18	0.15	0.19	0.16	0.14
Average Separation (Mean)	11.61	11.65	11.70	11.77	11.88	11.89

these are very big differences noticeable by eye when plotting the data, therefore reducing the number of mean-shifted variables from 244 to 3 using the MAXDEV method results in the size of the offsets required for separation of the two data sets being not acceptable.

### Case MS1

The mean-shifting procedure for case MS1, with S500 and offset 40.45 (corresponding to a misclassification rate of  $\approx 0\%$ ), can be seen in Figure 8.21. The variable selected for illustration of the mean-shifting was at 5.78 *ppm* with standard deviation 0.876 (Table A.1). Experiments using offsets in the range 4.48 – 49.40 in all three sample size



**Figure 8.21:** Illustration of the mean-shifting procedure in the case MS1 with S500 and offset 40.45 (MAXDEV). The brown and blue lines are the mean spectra of the reference and test data set respectively. The mean-shifted variables correspond to the points in the spectra for which the blue line is above the brown. The mean-shifted variable is at 5.78 *ppm*.

cases showed that the two data sets are linearly separated for offsets above 40.45, with LDA misclassification rates below 1% and average separation value of  $\approx 12$ . Table 8.11 shows the results of the experiments in the cases S100, S500 and S1000 respectively, for offsets in the range 4.48 – 49.40. From Table 8.11, it can be seen that offsets in the range 4.48 – 40.45 are required in all sample size cases, to achieve misclassification rates of  $\approx 25 - 0.5\%$  respectively. The average separation between the two data sets is  $\approx 11.3$  for a 20% misclassification rate and the two data sets are almost linearly separable when the average separation is  $\approx 12$ , with misclassification rate less than 1% as expected. In general, there are no great differences among the three sample size cases with respect to the *misclassification rate*, *average separation* and offsets required to obtain those statistics. However, there are noticeable differences in the required offsets from the previous cases, as in case MS1 they are much higher than for the rest of the cases.

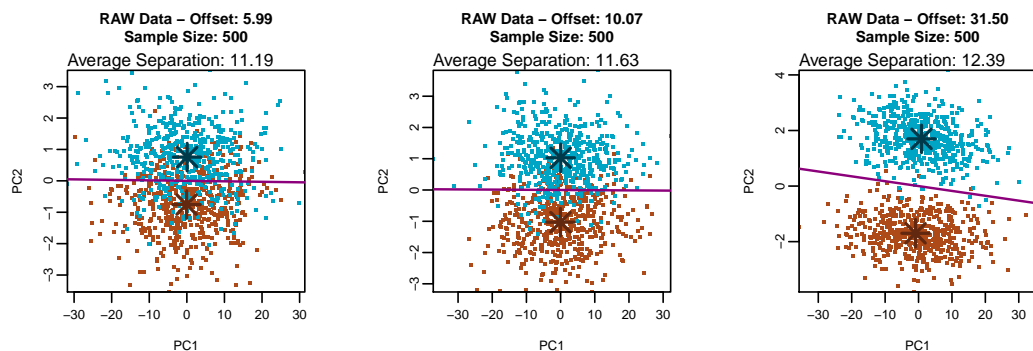
An illustration of how the mean-shifting procedure affects the capability of PCA to discriminate between the two data sets in this case can be seen in Figure 8.22, for offsets 5.99, 10.07 and 31.50 which correspond to 20%, 10% and 1% misclassification rates respectively.

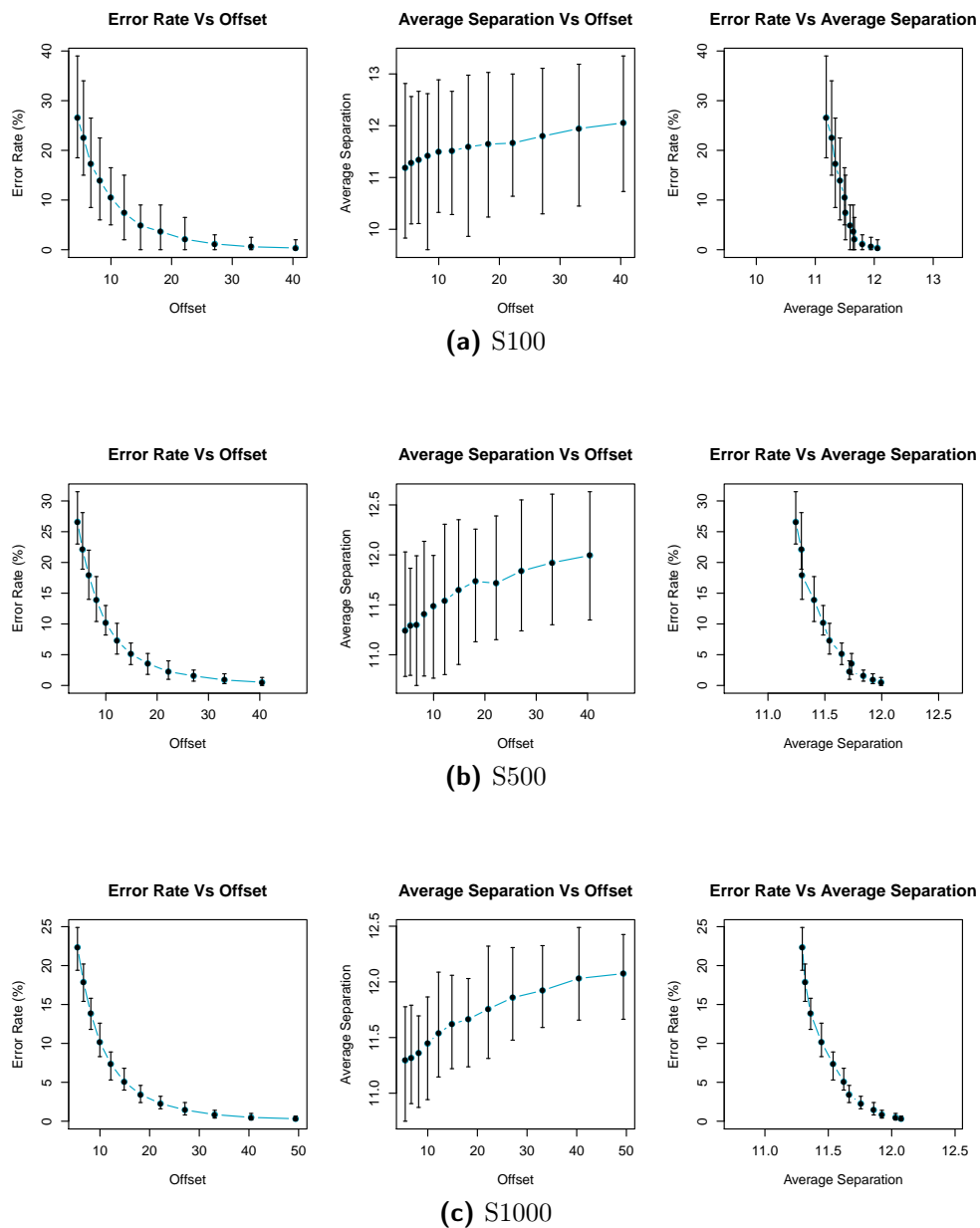
The relation among *misclassification rates*, *average separation* and offsets in the case MS1 in 100 runs, is shown in Figure 8.23. In the case MS1 with MAXDEV, average

**Table 8.11:** Average LDA misclassification rates and average separation values for the case MS1, applying the MAXDEV method in 100 runs of the experiment.

<b>S100</b>						
<b>Offset</b>	4.48	5.47	6.69	8.17	9.97	12.18
<b>Error Rate (%)</b>	26.59	22.53	17.25	13.90	10.51	7.40
<b>Average Separation</b>	11.18	11.27	11.33	11.41	11.49	11.51
<b>Offset</b>	14.88	18.17	22.20	27.11	33.12	40.45
<b>Error Rate (%)</b>	4.87	3.64	2.11	1.16	0.59	0.35
<b>Average Separation</b>	11.59	11.64	11.66	11.79	11.94	12.05
<b>S500</b>						
<b>Offset</b>	4.48	5.47	6.69	8.17	9.97	12.18
<b>Error Rate (%)</b>	26.59	22.08	17.89	13.89	10.18	7.31
<b>Average Separation</b>	11.24	11.29	11.29	11.40	11.48	11.54
<b>Offset</b>	14.88	18.17	22.20	27.11	33.12	40.45
<b>Error Rate (%)</b>	5.11	3.56	2.25	1.54	0.88	0.52
<b>Average Separation</b>	11.64	11.73	11.71	11.83	11.92	11.99
<b>S1000</b>						
<b>Offset</b>	5.47	6.69	8.17	9.97	12.18	14.88
<b>Error Rate (%)</b>	22.34	17.88	13.82	10.16	7.36	5.07
<b>Average Separation</b>	11.29	11.31	11.36	11.44	11.53	11.62
<b>Offset</b>	18.17	22.20	27.11	33.12	40.45	49.40
<b>Error Rate (%)</b>	3.42	2.26	1.44	0.84	0.48	0.31
<b>Average Separation</b>	11.66	11.75	11.85	11.92	12.03	12.07

separation values are in the interval 9.5 – 13.1, for LDA misclassification rates between 0 and 40 %, similarly to the MS3 case. As in the previous cases, the sample size does not play any role in the offsets in case MS1, as the required offsets to achieve misclassification rates of 0 – 20 % are similar in all three sample size cases. The range of values of the two statistics obtained in 100 runs reduces significantly as the sample

**Figure 8.22:** Visualisation of the LDA boundaries for the two artificial data sets in the case MS1 (MAXDEV). The data corresponds to the first two PCs for LDA. The reference and test data points are depicted in brown and blue respectively.



**Figure 8.23:** Graphical representation of the relation among *LDA misclassification rates*, *average separation* and *offsets* in the case MS1 applying the MAXDEV method. The blue lines represent the mean values of each statistic for each offset in the selected offset range. The offsets are the multiplicative factors on the original scale of the data. The vertical error bars are such that the top and bottom of a bar correspond to the maximum and minimum statistic value at the respective offset. The statistics values are the average values in 100 runs of the experiment. The two average separation plots in each sample size case are drawn using the same range of values for the *average separation*.

size increases, indicating that the stability of the misclassification rates depends on the sample size.

Table 8.12 gives the results for the CV, standard deviation and mean of the error bars for the two statistics in case MS1 for all three sample sizes. The CV of the

**Table 8.12:** Coefficient of variation results for case MS1 using method MAXDEV, of the *LDA misclassification rates and average separation* values in 100 runs of the experiment.

S100						
Offset	4.48	5.47	6.69	8.17	9.97	12.18
Error Rate (CV)	15.38	14.94	19.94	23.87	21.32	31.18
Error Rate (StDev)	4.09	3.37	3.44	3.32	2.24	2.31
Error Rate (Mean)	26.59	22.53	17.25	13.90	10.51	7.40
Average Separation (CV)	5.23	4.97	4.64	5.02	4.69	4.38
Average Separation (StDev)	0.59	0.56	0.53	0.57	0.54	0.50
Average Separation (Mean)	11.19	11.28	11.34	11.42	11.50	11.51
Offset	14.88	18.17	22.20	27.11	33.12	40.45
Error Rate (CV)	36.34	41.74	59.62	73.72	119.47	142.86
Error Rate (StDev)	1.77	1.52	1.26	0.86	0.70	0.50
Error Rate (Mean)	4.88	3.64	2.12	1.17	0.59	0.35
Average Separation (CV)	5.83	4.45	4.09	4.65	3.90	4.51
Average Separation (StDev)	0.68	0.52	0.48	0.55	0.47	0.54
Average Separation (Mean)	11.59	11.64	11.66	11.80	11.94	12.06
S500						
Offset	4.48	5.47	6.69	8.17	9.97	12.18
Error Rate (CV)	6.45	7.55	8.61	8.35	10.55	14.28
Error Rate (StDev)	1.72	1.67	1.54	1.16	1.07	1.04
Error Rate (Mean)	26.59	22.09	17.89	13.89	10.19	7.31
Average Separation (CV)	2.17	1.93	2.08	1.91	2.16	2.36
Average Separation (StDev)	0.24	0.22	0.23	0.22	0.25	0.27
Average Separation (Mean)	11.24	11.29	11.30	11.41	11.49	11.54
Offset	14.88	18.17	22.20	27.11	33.12	40.45
Error Rate (CV)	15.83	17.54	25.12	26.68	36.28	44.77
Error Rate (StDev)	0.81	0.63	0.57	0.41	0.32	0.23
Error Rate (Mean)	5.11	3.56	2.25	1.55	0.89	0.52
Average Separation (CV)	2.23	1.97	2.17	2.10	2.21	2.16
Average Separation (StDev)	0.26	0.23	0.25	0.25	0.26	0.26
Average Separation (Mean)	11.65	11.74	11.72	11.84	11.92	11.99
S1000						
Offset	5.47	6.69	8.17	9.97	12.18	14.88
Error Rate (CV)	4.90	5.27	5.95	7.48	10.47	10.45
Error Rate (StDev)	1.10	0.94	0.82	0.76	0.77	0.53
Error Rate (Mean)	22.35	17.89	13.83	10.16	7.36	5.08
Average Separation (CV)	1.61	1.53	1.46	1.48	1.45	1.25
Average Separation (StDev)	0.18	0.17	0.17	0.17	0.17	0.15
Average Separation (Mean)	11.29	11.32	11.36	11.45	11.54	11.62
Offset	18.17	22.20	27.11	33.12	40.45	49.40
Error Rate (CV)	12.42	15.21	20.85	26.93	32.93	37.49
Error Rate (StDev)	0.43	0.35	0.30	0.23	0.16	0.12
Error Rate (Mean)	3.43	2.27	1.45	0.85	0.49	0.31
Average Separation (CV)	1.51	1.59	1.54	1.47	1.38	1.35
Average Separation (StDev)	0.18	0.19	0.18	0.17	0.17	0.16
Average Separation (Mean)	11.66	11.76	11.86	11.92	12.03	12.07

*misclassification rate* increases as the offsets increase, with the sample size affecting the

CV values such that, independently of the offsets, the larger the sample size is, the lower the CV values are. The most consistent behaviour (monotonic increase) of the CV values is observed in the S1000 case. The CV values for the *average separation* are consistent for all three sample sizes, being in general quite similar to those of the other MS cases using MAXDEV, but lower than those of MS244. Their pattern is inconsistent with respect to decreasing or increasing, independently of the sample size, which shows that they depend only slightly (if at all), on the offsets. Thus, the CV of the error bars for both statistics are affected by the sample size, but only the *misclassification rate* depends heavily on the offsets. In addition, the error bars of the *misclassification rate* depend more heavily on the number of mean-shifted variables than the *average separation*, as the 5 CV tables for all MS cases seen so far indicate. Therefore, the findings of MS3 are confirmed by those of MS1 for the CV values of the two statistics.

PCA can discriminate between the two data sets with a misclassification rate of 1% or less, if there is an approximate thirty-fold increase in the mean of the variable with the highest standard deviation. This is not acceptable, as in the MS3 case, therefore reducing the number of mean-shifted variables from 244 to 1 using the MAXDEV method results in data sets that cannot be linearly separated. In other words, small changes in only one variable are unlikely to be detected through using PCA only.

A summary of the *misclassification rate* percentages with regard to the required offsets to achieve those rates for all four variable cases can be seen in Table 8.13. This

**Table 8.13:** Summary results (offsets) for the *LDA misclassification rates* in all MS cases for MAXDEV. The results are for 100 runs of the simulation algorithm and the offsets correspond to multiplicative factors on the original scale of the data. An offset is the value above which a selected misclassification rate percentage is positively achieved, e.g. in case MS20 with S500 at most 10% of the samples are misclassified when the offset is 1.61 or above.

Subset of Variables	Sample Size	Misclassification Rate				
		20%	15%	10%	5%	1%
MS120	S100	1.27	1.29	1.32	1.36	1.44
	S500	1.27	1.29	1.32	1.37	1.44
	S1000	1.27	1.29	1.32	1.37	1.44
MS20	S100	1.52	1.56	1.61	1.69	1.85
	S500	1.52	1.56	1.61	1.69	1.85
	S1000	1.52	1.56	1.61	1.69	1.85
MS3	S100	2.82	3.18	3.78	4.66	6.61
	S500	2.82	3.22	3.74	4.66	6.68
	S1000	2.82	3.22	3.74	4.66	6.68
MS1	S100	5.92	7.61	10.07	15.02	28.50
	S500	5.98	7.61	10.07	15.02	31.50
	S1000	5.98	7.61	10.07	15.02	31.50



summary table shows that for PCA to discriminate between the two data sets with a misclassification rate of 1% or less requires mean-shifting between 20 and 200 variables, as in case MS20 the increase of the selected variables' means is 100% and in case MS120 50%. Mean-shifting subsets of variables of number less than 20 increases significantly the required offsets to a practically infeasible size, e.g. a seven-fold increase in the case MS3, a thirty-fold increase in the case MS1 and a sixteen-fold increase in MS244. The sample size is not important in any of the cases, as there are no differences among the results for S100, S500 and S1000 in all cases for MAXDEV. The *average separation* increases as the number of mean-shifted variables increases, as the results of the five MS cases indicate.

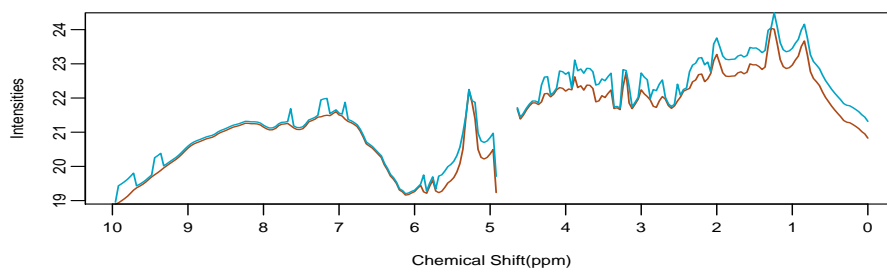
## 8.4.4 Minimum Deviation (MINDEV)

### Introduction

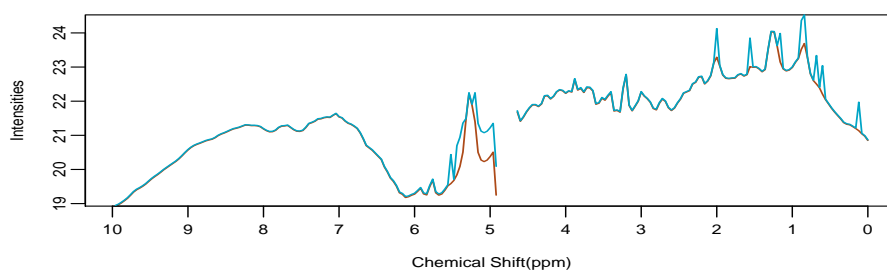
As described in Subsection 8.4.1, here the main findings are stated briefly, and the results are given in the appendices. Experiments performed in the case MAXDEV showed that in the cases MS3 and MS1, independently of the sample size, PCA cannot discriminate between the two data sets and the LDA algorithm cannot separate linearly the two data sets, as the required offsets for an LDA misclassification rate of less than 1% are larger than 7 and 28.5 in cases MS3 and MS1 respectively. In this section, another method, MINDEV, for selecting the variables to mean-shift is used in the simulation experiments. This method selects a specific subset of variables for mean-shifting according to increasing order of size of their standard deviation. Similarly to MAXDEV, it will be applied to the simulation experiments for subsets of 120, 20, 3 and 1 variables, and for all three sample size cases, S100, S500 and S1000. The lists of mean-shifted variables and their standard deviation for all MS cases can be seen in Table A.2 in increasing order of standard deviation. The three variables with the smallest standard deviations in increasing order are at 4.98, 4.94 and 5.02 ppm with standard deviations 0.584, 0.585 and 0.592 respectively.

### MINDEV Simulation Results

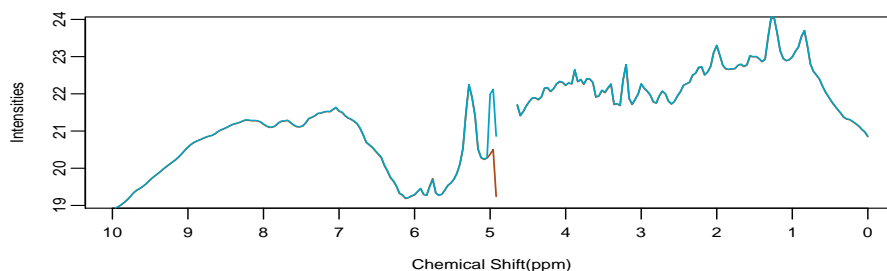
Figure 8.24 illustrates the mean-shifting effect in the four MS cases for offsets 1.55, 2.27 4.95 and 12.18 respectively. In general, the sample size does not play any role in the results of the four MS cases, as the offsets to achieve 1 – 20 % misclassification rates are quite similar for all three sample sizes (Tables B.1 - B.4). However, it is clear that the number of mean-shifted variables is important for the capability of PCA to discriminate between the two sets, since for instance an offset of 1.44 is required to achieve a misclassification rate of 1 % in case MS120, whereas in case MS1 the same misclassification rate is at an offset of approximately 10. As Figures B.2 - B.5 show, the



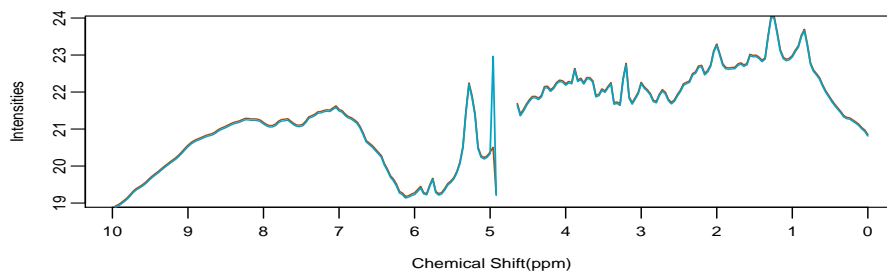
(a) Case: MS120 - Offset: 1.55



(b) Case: MS20 - Offset: 2.27



(c) Case: MS3 - Offset: 4.95



(d) Case: MS1 - Offset: 12.18

**Figure 8.24:** Illustration of the mean-shifting procedure for MINDEV in all MS cases with S500. The brown and blue lines are the mean spectra of the reference and test data set respectively. The mean-shifted variables correspond to the points in the spectra for which the blue line is above the brown. The offsets are such that the misclassification rate is  $\approx 0\%$  in all four MS cases.

two statistics seem to be increasingly less dispersed when the number of mean-shifted variables decreases.

Tables B.5 -B.8 indicate that the coefficient of variation for the two statistics depends on the sample size, but not on the number of mean-shifted variables. However, only the CV of the *misclassification rate* depends on the offsets. The dispersion (CV) of the *average separation* is much smaller and less dependent on the number of mean-shifted variables than that of the *misclassification rate* for all MS cases and sample sizes. Considering case MS244, MAXDEV and MINDEV, the CVs of the two statistics obtained in all MS cases using MINDEV are smaller in case MS244 than for MAXDEV and MINDEV, although in the case of *average separation*, for small offsets in all MS cases using MAXDEV and MINDEV, the CVs of the statistics are smaller than those of MS244.

Figure B.1 illustrates the PC scores plots superimposed with the LDA boundaries in the four MS cases for 20 %, 10 % and 1 % misclassification rates with sample size S500.

A summary of the *misclassification rate* percentages with regard to the required offsets to achieve those rates for the four variable cases, MS120, MS20, MS3 and MS1, can be seen in Table 8.14. This table shows that in the case of MINDEV, independently of the sample size, it is feasible for PCA to discriminate between the two data sets in the cases MS120 and MS20, at 1% error rate, with an approximately 44 % and 85 % increase of the means of the selected variables. In addition, an increase of the selected variables' means of approximately 400 % (four-fold) and 1000 % (ten-fold) in cases MS3 and MS1 respectively is required, which is clearly not feasible. Therefore, PCA can discriminate between the two data sets in the case of MINDEV, only if at least 10 % of the variables in the data set are mean-shifted.

## 8.4.5 Maximum Mean (MAXMEAN)

### Introduction

Previous results showed that at least 10 % of the variables need to be mean-shifted in order to allow PCA to discriminate between the two data sets. This section covers the simulation experiments using MAXMEAN. That is, a specific subset of variables to mean-shift is chosen according to decreasing order of their size of mean. This will be applied in the usual MS and S cases seen so far. The lists of mean-shifted variables and their standard deviations for all MS cases can be seen in Table A.3 in decreasing order of their mean. The three variables with the largest means in decreasing order are 1.30, 1.26 and 0.86 with means 24.05, 24.04 and 23.70 respectively.

### MAXMEAN Simulation Results

The mean-shifting procedure for the four MS cases, with S500 and offsets 1.55, 2.27, 4.95 and 16.50 (corresponding to a misclassification rate of  $\approx 0\%$ ), can be seen in Figure 8.25.

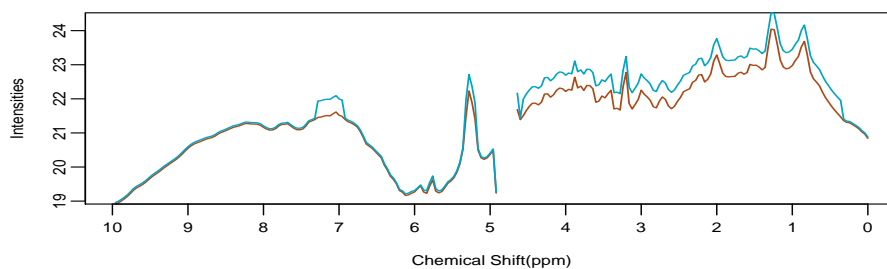
Tables B.9 - B.12 show that, similarly to the previous two methods, only the number of variables to mean-shift plays any role in the offsets required for linear separation of the two data sets. The average separation values are affected, as the smaller the number of mean-shifted variables is, the smaller the value of average separation is in an experiment, e.g. the average separation values in cases MS120 and MS1, with S500 and misclassification rate 1 % are 11.89 and 11.38 respectively. The PC scores plots and the LDA boundaries in the four MS cases for 20 %, 10 % and 1 % misclassification rates with S500 can be seen in Figure B.6. Higher instability in the values of the two statistics is observed in MS cases of smaller numbers of variables. Especially in the case of the *misclassification rate* this fact is more evident than in the *average separation* (Figures B.7 - B.10).

The coefficient of variation for the two statistics, as can be seen in Tables B.13 - B.16, confirms that the *average separation* is more stable than the *misclassification rate* in all three sample size cases. Both statistics are affected by the sample size in all MS cases. As with the previous cases, MS244, MAXDEV and MINDEV, the CV of the *misclassification rate* is clearly affected by the offsets in all MS cases using MAXMEAN, whereas the CV of the *average separation* is independent of the offsets, for all MS cases and sample sizes. The dispersion of *average separation* is consistently much smaller and is affected far less by the number of mean-shifted variables, than that of the *misclassification rate*. In general, the CV of the *misclassification rate* is larger for MAXDEV and MINDEV, but smaller in case MS244, than that of MAXMEAN, while in the case of *average separation*, there is practically no difference in its value, between the methods used to select the variables to mean-shift.

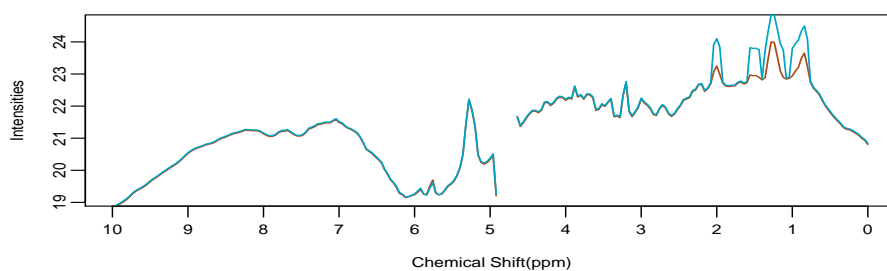
In summary, from Table 8.14 it can be seen that similarly to the previous methods, using MAXMEAN, at least 10 % of the variables in the data set need to be mean-shifted to achieve linear separation of the two data sets, independently of the sample size. More specifically, PCA can discriminate between the two sets in cases MS120 and MS20 with an approximate 44 % and 85 % increase, respectively, of the means of the variables selected by MAXMEAN (which is feasible), whereas a four-fold and eleven-fold increase is required in the cases MS3 and MS1 respectively (which is not practically feasible).

## 8.5 Conclusions

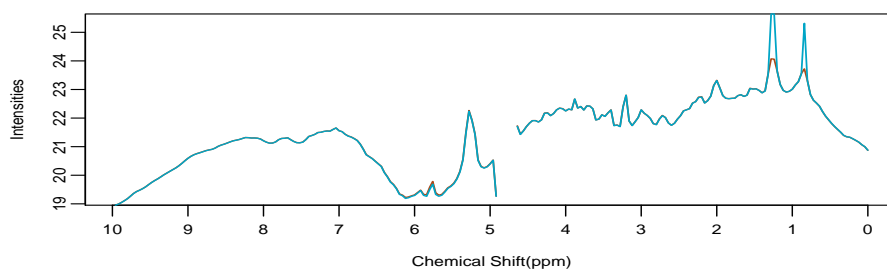
In the previous sections of this chapter a series of simulation experiments was performed to investigate the capability of PCA to discriminate between two groups of points,



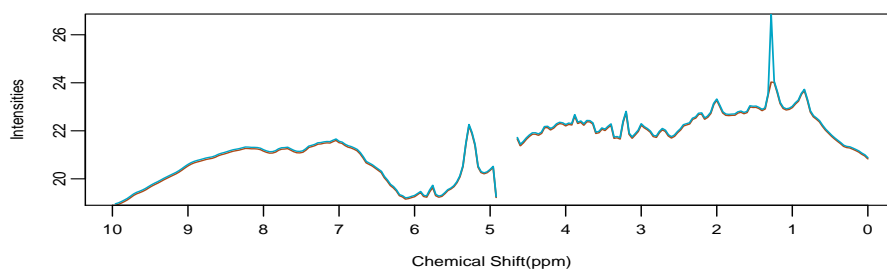
(a) Case: MS120 - Offset: 1.55



(b) Case: MS20 - Offset: 2.27



(c) Case: MS3 - Offset: 4.95



(d) Case: MS1 - Offset: 16.50

**Figure 8.25:** Illustration of the mean-shifting procedure for MAXMEAN in all MS cases with S500. The brown and blue lines are the mean spectra of the reference and test data set respectively. The mean-shifted variables correspond to the points in the spectra for which the blue line is above the brown. The offsets are such that the misclassification rate is  $\approx 0\%$  in all four MS cases.

a reference and a test data set. Experiments included the mean-shifting of selected subsets of variables, comprised of 244, 120, 20, 3 and 1 variables, using three different methods of selecting these variables, namely MAXDEV, MINDEV and MAXMEAN. In addition, three sample size cases were examined, S100, S500 and S1000, to investigate the possibility of the sample size of the two artificially generated data sets playing any role in the discriminating ability of PCA. Two different statistics were used to assess the discriminating performance of the simulation algorithm, the *LDA misclassification rate* and the *average separation* of two distributions of points. A summary of the results obtained from all 39 experiments for all subsets of variables, sample sizes and selection methods can be seen in Tables 8.14 and 8.15 for the required offsets and the *average separation* of the two data sets with the corresponding offsets respectively.

From Table 8.14 and the results in the previous sections, the following can be concluded:

- The discriminating ability of PCA depends on the number of variables that are mean-shifted. In the cases MS120 and MS20, the required offsets for linear separability of the two artificial data sets are less than 2, meaning that the increase in the means of the selected variables is at most two-fold, and such differences may occur in practice. In general, for any subset of variables of size above 120 or below 20, the required offsets are between  $\approx 3.93$  and 31.50, which are clearly not feasible, as the structure of the test data set is then no longer similar to the structure of the original epilepsy data. This is confirmed in cases MS244, MS3 and MS1 where PCA cannot discriminate between the two data sets which do retain the original epilepsy data structure.
- The sample size of the two data sets does not affect the results of the simulation experiments concerning the *misclassification rate*, as the offsets are similar in all MS cases for the three sample size cases chosen (S100, S500 and S1000). Sample size affects only the stability of the *misclassification rate*, as indicated by the misclassification rate vs offsets plots in all the experiments, since the range of values of the misclassification rate in all MS cases is considerably smaller in cases S500 and S1000 than in case S100 in 100 runs of the experiments. This means that in metabonomics studies the sample size does not really play any important role in the results, therefore it is of little value to collect data on 500 people as opposed to 200, unless subgroup analysis is required.
- Concerning the three methods used to select the variables, MAXDEV, MINDEV and MAXMEAN, differences in the offsets were observed only in experiments involving the cases MS3 and MS1. More specifically, in the cases MS244, MS120 and MS20, the offsets were not affected by the method used, as in all these experiments and independently of the sample size of the two data sets, they are similar for all five levels of misclassification rate (20, 15, 10, 5 and 1%). However,

in the case MS3 and even more in MS1, it is clear that the way the variables were selected affects the offsets required to achieve the misclassification rates in Table 8.14. That is, method MINDEV results in the smallest offsets among the three methods used in the experiments, which is not surprising as the variation using this method is smaller. The differences in offsets are more noticeable for misclassification rates of 10% and below. As expected due to the largest amount of variation, the worst method in cases MS3 and MS1 is MAXDEV (resulting in considerably larger offsets, especially in case MS1, than the other two methods), while the offsets for MAXMEAN are quite close to those of MINDEV. It should be noted though, that these three methods were only convenient ways of selecting the variables and do not have any medical or physical basis for their choice. That is, a random choice could also have been made instead. From the results for the three methods used to select the variables, it can be said that in MS cases with less than 20 mean-shifted variables, the lack of standard deviation in the variables, affects the offsets, being the smallest among all three methods. That is, in MINDEV, the fewer the mean-shifted variables are, the smallest the offsets required to achieve the selected levels of misclassification rate are. Using MAXMEAN results in these MS cases with offsets being approximately the average value of the offsets in the other two methods. That can be explained, since the subset of mean-shifted variables obtained by using MAXMEAN, contains both variables with high and medium or low standard deviation, as it can be confirmed by Tables A.1, A.2 and A.3. In general, large standard deviation in the variables negatively affects the offsets, as the results of the MAXDEV experiments indicate.

From Table 8.15 and the results in the previous sections, the following can be concluded:

- In general, the *average separation* statistic is very stable in the simulation experiments, as there are no great differences in its values among the various variable cases, sample size cases and methods used to select the variables. In Table 8.15, case MS244 is measured on a different scale than the rest of the cases. In the other four MS cases the average separation values are fairly consistent, in the range 11.19 – 11.91, for offsets in the range 20 – 1% respectively, independently of the sample size. As expected, the smaller the *misclassification rate* is, the higher the *average separation* of the two data sets is. However from the average separation vs offset plots, the increase of the *average separation* is smooth only in the case MS244, whereas in the rest of the MS cases it does not increase smoothly, as the graphical representation of the relation between *average separation* and offsets is not a curve and does not monotonically increase for each offset.
- The coefficient of variation of the *average separation* for each offset in 100 runs is far smaller than that of the *misclassification rate* and it is affected by the sample size of the two data sets. More specifically, the higher the sample size

is, the smaller the width of error bars variation (and the coefficient of variation) of the *average separation* at each offset is, therefore the more stable the *average separation* is. The last fact is also true for the *misclassification rate*, in which the CV is also affected considerably by the sample size.

### Row-scaling the Data

As was discussed in Section 8.3, it is possible to add an extra step, namely the row-scaling to a constant total of the original epilepsy data matrix, in the pre-processing of the original epilepsy data. This step would be carried out before log-transforming the elements of the epilepsy data matrix. As all the values of the data matrix after row-scaling are in the range 0 – 1, log-transforming the data causes all these values to be negative. This does not affect the execution of the simulation algorithm and the results of the analyses for all MS cases, apart from MS244, for which it is not appropriate to row-scale the data (an illustration of this can be seen in Figure 8.5), as seen in Table 8.16.

Comparing the results of the simulation experiments in Tables 8.14 and 8.16, it is clear that although the row-scaled data results are slightly better (in most cases the required offsets are slightly smaller than in the raw data), they are not small enough to be considered to improve the situation with cases MS3 and MS1. For example, in the case MS1 with S500, an offset of 24.05 is required to achieve 1% LDA misclassification rate in the row-scaled data, whereas in the raw data the same experiment requires an offset of 31.50. Despite the row-scaled offset being smaller than the raw offset, it is still very large and not feasible in practice. As this situation is similar in almost all the simulation experiments, the row-scaled data does not offer a clear improvement of the results to those of the raw data, and therefore it is not a preferred pre-treatment step for the raw data.

The simulation experiments in this chapter showed that to design an experiment based on PCA as the tool for discrimination of the data groups, large samples are not necessary, and that PCA will not be very useful unless there are a lot of variables changing in the ways mentioned previously. Other unsupervised techniques such as SOM could be more capable of discriminating between the groups if only a few variables change. With regard to the implications for the design and analysis of the epilepsy data that was used in the experiments,

- it is clear that there are not enough non-responder patients
- no differences were detected in a large number of variables, suggesting that here are no major differences between responders and non-responders.



**Table 8.14:** Summary results for the *offsets* required to achieve 20%, 15%, 10%, 5% and 1% misclassification rate for all variable methods, MS and S cases. The results are over 100 runs of the simulation algorithm and on the original scale of the data. An offset is the value above which a selected misclassification rate percentage is positively achieved, e.g in the case MS20 with S100 and MAXDEV, at most 10% of the samples are misclassified when the offset is 1.61 or above.

Subset of Variables	Sample Size	Selection Method	Misclassification Rate				
			20%	15%	10%	5%	1%
MS244	S100		2.85	3.49	4.71	7.61	15.64
	S500		2.85	3.56	4.75	7.69	16.11
	S1000		2.85	3.56	4.80	7.69	16.60
MS120	S100	MAXDEV	1.27	1.29	1.32	1.36	1.44
		MINDEV	1.25	1.28	1.30	1.36	1.43
		MAXMEAN	1.25	1.28	1.30	1.36	1.43
	S500	MAXDEV	1.27	1.29	1.32	1.37	1.44
		MINDEV	1.25	1.28	1.32	1.36	1.44
		MAXMEAN	1.25	1.28	1.32	1.36	1.44
	S1000	MAXDEV	1.27	1.29	1.32	1.37	1.44
		MINDEV	1.25	1.28	1.32	1.36	1.44
		MAXMEAN	1.25	1.28	1.32	1.36	1.44
MS20	S100	MAXDEV	1.52	1.56	1.61	1.69	1.85
		MINDEV	1.53	1.58	1.63	1.71	1.85
		MAXMEAN	1.52	1.56	1.61	1.69	1.84
	S500	MAXDEV	1.52	1.56	1.61	1.69	1.85
		MINDEV	1.53	1.58	1.63	1.71	1.85
		MAXMEAN	1.53	1.56	1.61	1.69	1.85
	S1000	MAXDEV	1.52	1.56	1.61	1.69	1.85
		MINDEV	1.53	1.58	1.63	1.71	1.85
		MAXMEAN	1.52	1.56	1.61	1.69	1.85
MS3	S100	MAXDEV	2.82	3.18	3.78	4.66	6.61
		MINDEV	3.00	3.12	3.25	3.49	3.93
		MAXMEAN	2.88	3.18	3.22	3.49	4.17
	S500	MAXDEV	2.82	3.22	3.74	4.66	6.68
		MINDEV	3.00	3.12	3.25	3.49	3.93
		MAXMEAN	2.88	3.15	3.18	3.49	4.17
	S1000	MAXDEV	2.82	3.22	3.74	4.66	6.68
		MINDEV	3.00	3.12	3.25	3.49	3.93
		MAXMEAN	2.88	3.00	3.18	3.45	4.17
MS1	S100	MAXDEV	5.92	7.61	10.07	15.02	28.50
		MINDEV	7.02	7.38	7.76	8.41	9.97
		MAXMEAN	6.35	6.82	7.38	8.58	11.02
	S500	MAXDEV	5.98	7.61	10.07	15.02	31.50
		MINDEV	7.09	7.38	7.61	8.16	10.07
		MAXMEAN	6.17	6.61	7.09	8.24	11.02
	S1000	MAXDEV	5.98	7.61	10.07	15.02	31.50
		MINDEV	7.09	7.38	7.69	8.08	9.02
		MAXMEAN	6.17	6.55	7.09	8.16	11.02

**Table 8.15:** Summary results for the *average separation* of the two data sets when the misclassification rate is 20%, 15%, 10%, 5% and 1% for all variable methods, MS and S cases. The results are over 100 runs of the simulation algorithm.

Subset of Variables	Sample Size	Selection Method	Misclassification Rate				
			20%	15%	10%	5%	1%
MS244	S100		17.89	20.98	25.06	31.13	43.97
	S500		18.04	21.03	25.22	31.69	45.27
	S1000		17.97	20.95	25.11	32.89	45.23
MS120	S100	MAXDEV	11.38	11.40	11.57	11.65	11.80
		MINDEV	11.33	11.46	11.52	11.69	11.86
		MAXMEAN	11.42	11.41	11.55	11.67	11.87
	S500	MAXDEV	11.32	11.44	11.53	11.69	11.91
		MINDEV	11.37	11.45	11.50	11.67	11.89
		MAXMEAN	11.38	11.46	11.51	11.62	11.88
	S1000	MAXDEV	11.34	11.43	11.54	11.62	11.89
		MINDEV	11.38	11.44	11.50	11.64	11.88
		MAXMEAN	11.38	11.42	11.50	11.62	11.89
MS20	S100	MAXDEV	11.27	11.33	11.39	11.45	11.54
		MINDEV	11.22	11.31	11.38	11.43	11.58
		MAXMEAN	11.48	11.23	11.35	11.40	11.55
	S500	MAXDEV	11.29	11.30	11.35	11.43	11.59
		MINDEV	11.26	11.28	11.35	11.45	11.59
		MAXMEAN	11.28	11.30	11.34	11.42	11.56
	S1000	MAXDEV	11.23	11.29	11.35	11.45	11.64
		MINDEV	11.29	11.32	11.34	11.46	11.59
		MAXMEAN	11.26	11.29	11.36	11.42	11.58
MS3	S100	MAXDEV	11.29	11.38	11.47	11.56	11.86
		MINDEV	11.24	11.28	11.27	11.36	11.45
		MAXMEAN	11.24	11.33	11.34	11.41	11.48
	S500	MAXDEV	11.31	11.34	11.45	11.58	11.87
		MINDEV	11.23	11.29	11.36	11.38	11.48
		MAXMEAN	11.24	11.27	11.35	11.37	11.51
	S1000	MAXDEV	11.29	11.38	11.46	11.60	11.87
		MINDEV	11.23	11.26	11.32	11.37	11.46
		MAXMEAN	11.23	11.28	11.30	11.38	11.48
MS1	S100	MAXDEV	11.30	11.39	11.49	11.58	11.82
		MINDEV	11.19	11.25	11.34	11.32	11.41
		MAXMEAN	11.23	11.23	11.36	11.41	11.49
	S500	MAXDEV	11.29	11.38	11.48	11.64	11.90
		MINDEV	11.25	11.29	11.28	11.31	11.38
		MAXMEAN	11.26	11.24	11.26	11.37	11.42
	S1000	MAXDEV	11.30	11.34	11.45	11.62	11.90
		MINDEV	11.26	11.28	11.33	11.32	11.37
		MAXMEAN	11.24	11.27	11.31	11.33	11.47

**Table 8.16:** Summary results for the *offsets* required to achieve 20%, 15%, 10%, 5% and 1% misclassification rate for all variable methods, MS and S cases. The results are over 100 runs of the simulation algorithm and on the original scale of the data. The data are ROW-SCALED and then LOG-TRANSFORMED. An offset is the value above which a selected misclassification rate percentage is positively achieved, e.g. in the case MS120 with S1000 and MINDEV, at most 5% of the samples are misclassified when the offset is 1.29 or above.

Subset of Variables	Sample Size	Selection Method	Misclassification Rate				
			20%	15%	10%	5%	1%
MS120	S100	MAXDEV	1.23	1.25	1.26	1.30	1.36
		MINDEV	1.22	1.24	1.26	1.29	1.35
		MAXMEAN	1.21	1.23	1.27	1.31	1.40
	S500	MAXDEV	1.23	1.25	1.27	1.30	1.36
		MINDEV	1.22	1.25	1.26	1.28	1.34
		MAXMEAN	1.22	1.25	1.28	1.32	1.40
	S1000	MAXDEV	1.23	1.25	1.27	1.30	1.36
		MINDEV	1.23	1.25	1.26	1.29	1.35
		MAXMEAN	1.22	1.25	1.28	1.33	1.41
MS20	S100	MAXDEV	1.43	1.52	1.62	1.75	2.01
		MINDEV	1.45	1.48	1.51	1.55	1.63
		MAXMEAN	1.34	1.39	1.47	1.57	1.74
	S500	MAXDEV	1.44	1.52	1.62	1.76	2.07
		MINDEV	1.46	1.49	1.52	1.55	1.62
		MAXMEAN	1.35	1.41	1.47	1.58	1.78
	S1000	MAXDEV	1.45	1.52	1.62	1.77	2.05
		MINDEV	1.46	1.49	1.52	1.55	1.62
		MAXMEAN	1.34	1.40	1.47	1.57	1.79
MS3	S100	MAXDEV	2.35	2.68	3.05	3.88	5.37
		MINDEV	2.64	2.71	2.81	2.91	3.21
		MAXMEAN	2.07	2.28	2.53	2.96	3.72
	S500	MAXDEV	2.37	2.68	3.15	3.88	5.85
		MINDEV	2.68	2.75	2.79	2.88	3.06
		MAXMEAN	2.09	2.31	2.56	3.02	4.16
	S1000	MAXDEV	2.37	2.68	3.16	3.89	5.85
		MINDEV	2.69	2.74	2.80	2.88	3.06
		MAXMEAN	2.09	2.30	2.56	3.01	4.16
MS1	S100	MAXDEV	5.47	6.81	8.88	12.61	22.30
		MINDEV	5.33	5.61	5.83	6.28	7.29
		MAXMEAN	3.78	4.41	5.30	6.85	8.94
	S500	MAXDEV	5.48	7.09	8.62	12.80	24.03
		MINDEV	5.46	5.66	5.80	6.14	6.82
		MAXMEAN	3.86	4.45	5.45	7.47	19.69
	S1000	MAXDEV	5.58	6.89	8.94	12.81	24.29
		MINDEV	5.47	5.70	5.75	6.11	6.69
		MAXMEAN	3.74	4.39	5.42	7.42	19.11

# Chapter 9

## Conclusions and Further Work

### 9.1 Conclusions

This thesis covers a range of unsupervised multivariate statistical techniques, with the aim of investigating the possibility that these methods can be useful in the exploratory analysis of metabonomics data. For this purpose, a data set containing 122 patients with epilepsy was used to compare the results of the analyses obtained by these techniques. A review of NMR and MS, two important analytical techniques in metabonomics, was the subject of Chapter 3, and detailed description of the available pre-processing and pre-treatment techniques is given in Chapter 4. It was hoped that detailed analysis of these data would allow the determination of blood serum metabolites which could discriminate the patients between responders and non-responders to AEDs. This analysis involved a novel comparison of a variety of statistical clustering techniques, and an assessment of their suitability for the analysis of such NMR data. Commonly used methods included PCA, HCA and  $k$ -means, whereas novel methods seldom used in this area included methods such as MDS, Fuzzy clustering and SOM. In general, all these methods were capable of identifying some structure in the epilepsy data, according to some patient clinical characteristics. Finally, a simulation investigation was carried out to assess the ability of PCA to identify structure under specific conditions such as data sets of different sample sizes, for all clinical characteristics of the patients.

The data used for the analyses was generated by proton NMR spectroscopy and before the application of any statistical method, the data was pre-processed (by an NMR researcher) and pre-treated in ways such as row and column-scaling, as well as element transformations of the data matrix. The original dimensions of the data (338) were reduced to 144 (in the range 0.02 – 5.98 ppm). Various scaling and transformation methods were investigated with the data at the end being row-scaled to a constant total and column-scaled by mean-centring of the variables. Table 9.1 gives the results of the comparison of two criteria, *Normalised Entropy* and the *Gleason - Staelin* statistic using the correlation matrix, both described in detail in Chapter 5, in assessing the

data suitability for PCA. It is clear that log-transforming the data affects negatively

**Table 9.1:** Comparison of scaling techniques using the original epilepsy data. The labels UNSCALED, ROW and LOG mean that the data are unscaled, are row-scaled to a constant total and are natural log-transformed, respectively. The data contain the variables in the spectral regions  $5.98 - 0.02$  ppm.

Data Set	Normalised Entropy	Gleason - Staelin
UNSCALED	0.064	0.953
ROW	0.121	0.555
LOG, ROW	0.349	0.559

(increasing) the value of the *Normalised Entropy*, whereas the value of the *Gleason - Staelin* statistic is approximately the same to that of the ROW data. Thus, the variables are considerably more correlated, and consequently more suitable for PCA, in the row-scaled data than in the log-transformed data. In addition, the percentage of the total variance explained by the first three PCs, for the three data sets mentioned previously, can be seen in Table 9.2. In the ROW case, only three PCs are required to explain

**Table 9.2:** Proportions of the total variance explained by the first three principal components. The numbers in parentheses () are the cumulative proportions. The labels UNSCALED, ROW and LOG mean that the data are unscaled, are row-scaled to a constant total and are natural log-transformed, respectively. The data contain the variables in the spectral regions  $5.98 - 0.02$  ppm.

Data Set	PC1	PC2	PC3
UNSCALED	93.30% (93.30%)	5.49% (98.79%)	0.68% (99.48%)
ROW	88.30% (88.30%)	6.64% (94.94%)	1.27% (96.22%)
LOG, ROW	43.55% (43.55%)	29.06% (72.61%)	13.40% (86.02%)

approximately 96% of the total variation in the data, while in the LOG case more than three PCs are clearly required to explain the same percentage of variation in the data. This makes the pattern recognition procedure more difficult, as at least a 4-dimensional space is required to describe the original data, which means that in the LOG case, the structure of the data cannot easily be visualised. Thus, from the results of the two tables, it is indicated that the ROW case should be used, but no log-transformation should take place. In addition, mean centring was applied to the columns of the data set.

Two exploratory data unsupervised techniques, *Principal components analysis* (PCA) and *Multi-dimensional scaling* (MDS) were used initially (the techniques and their results are described in detail in Chapters 5 and 6 respectively), to project the original input space of the data to a 2- or 3-dimensional output space (effectively reducing thus the required number of dimensions for pattern recognition) and to facilitate the identification of any patterns in the data. PCA is restricted to Euclidean spaces, but on the other hand it allows the investigation of any relationship between variables and samples.

MDS can be used with any dissimilarity (or similarity) measure, but it is very difficult to extract any information about the variables (or the spectral regions) from the results of the MDS analyses.

Results of the PCA were positive for four of the five clinical characteristics in question, i.e. *Gender*, *Seizure Type*, *Age* and *BMI*. It was shown that the first two or three PCs can separate the patients with respect to these characteristics. More importantly, concerning the fifth clinical characteristic, i.e. the *Response to AEDs*, no PC was capable of separating the patients into responders and non-responders. This fact was confirmed by constructing a general linear model with the first four PCs being the explanatory variables and the (recoded to 0-1 values) *Response to AEDs* information of the patients, as the dependent variable. Investigating the relationship between variables in the data, and between variables and samples, certain relations were found, so that patients belonging to specific categories of the clinical characteristics were considered to associate with specific variables in the NMR spectra. More specifically, it was discovered that, as PC2 indicated very high negative loadings for variables 1.26, 1.22, 0.86 and 3.22 and a high positive loading for 1.30, *females* usually have larger intensity values than *males* on these four variables, and smaller values than *males* on variable 1.30. Information about *Seizure Type* was conclusive only for those patients with *IGE* type, with variables 1.3 and 0.9 associated with them, so that these patients have larger intensity values on these two variables than the patients of *LRE* type do. Concerning the *Age* categories of the patients, those in category [26-47) have higher values of variables 1.3 and 1.34, while category [47-99) have higher values of variables 1.26, 1.22, 0.86 and 3.22. In addition, patient 44 could be related to the variables in the range 3.46–4.1 ppm. Finally, variables in the range 1.26 - 1.34 ppm and at 0.9 ppm seem to be associated with patients belonging to the two largest *BMI* categories, such that patients in these two categories have larger intensity values of these variables than those patients with *BMI* values in the two lower categories. In general, PCA has been proved useful to the pattern recognition of the data, with respect to the clinical characteristics of the patients, apart from the *Response to AEDs* for which no PCs were capable of identifying any patterns in the data.

The second dimension reduction and data visualisation technique that was used was the *classical* MDS method, and the derived 2-dimensional configuration was used as input to Sammon's *non-linear mapping* (NLM) method. An important advantage of these methods is that the required between-samples distances can be calculated using any dissimilarity (or similarity) measure. Comparing the MDS configurations derived by various distance metrics, it was shown that *classical* MDS was not capable of giving more information than PCA about the clustering behaviour of the patients with respect to their clinical characteristics. Results confirmed the findings of PCA for all clinical characteristics, including the *Response to AEDs*, for which none of the two

methods identified any clustering patterns. Two MDS configurations were retained, the *Euclidean* and the *Maximum* (both having far better results in identifying clustering patterns for the patients than any other distance metric examined), and were used for further analyses as input to NLM. It was found that the derived *Maximum* NLM configuration fitted the data slightly better than the *Euclidean* NLM configuration. In general, the *Maximum* NLM configuration proved to be slightly better than the two MDS and the *Euclidean* NLM configurations in separating the patients regarding their clinical characteristics, and, especially in the case of *Age* and *BMI*, it provided the best separation. Nevertheless, none of the four configurations, or any configuration for that matter, achieved any separation of the patients with regards to their *Response to AEDs*.

The data exploration part of the thesis gave good indications concerning the clustering patterns of the patients and their clinical characteristics. The next step was to apply the data clustering methods, in order to develop clustering methods confirming the findings of the exploratory stage of the pattern recognition analyses. Four different categories of unsupervised classification methods were assessed and clustering methods were developed for them. More specifically, considering the relevant literature for metabonomics data, *hierarchical clustering* algorithms, *optimal partitioning* methods and *competitive learning* algorithms were more or less the most popular and suitable unsupervised classification techniques, therefore they were chosen for the clustering analysis of the epilepsy data.

Hierarchical clustering algorithms such as *agglomerative nesting* algorithms are important in the area of metabonomics. These methods involve more than one step to establish the clustering patterns of the data, and each patient is assigned to one and only cluster. The data is clustered in a form of a dendrogram, showing the relationships between the patients. The procedure initially assigns one patient per cluster, and ends when all samples are contained in a single cluster. An important advantage of these algorithms is that the derived clusters are not restricted to a spherical shape, therefore, depending on the data in question, they might be more useful and flexible than other clustering methods. For example, *single linkage* produces non-compact elongated clusters, whereas *Ward's* method produces compact spherical clusters. Four distance measures were used to calculate the distances between the samples, and seven *agglomerative nesting* algorithms were used, such that 28 HCA methods were constructed and their results compared. Among these clustering methods, using suitable statistics and tests, the best method was found to be the 2-cluster partition derived by the *Maximum - Ward* HCA method. This method provided the best overall fit to the data, discriminating the patients regarding their clinical characteristics *Gender*, *Age* and *BMI*, but not in terms of their *Seizure Type* and *Response to AEDs*.

Optimal partitioning methods are non-hierarchical clustering algorithms, based on the minimisation of an objective function. In these algorithms, in general, the objects

are split into a predefined number of clusters, without any hierarchical relationship between partitions of different numbers of clusters, such that the items in a cluster are similar to each other, but different to those in other clusters. Two popular such methods, i.e. the *fuzzy* clustering and the *hard* clustering, were used in the thesis, for comparison purposes.

The main characteristic of fuzzy clustering algorithms is that an object can belong, to a certain degree, to more than one cluster at the same time. Such clustering algorithms can provide clusters of any shape, therefore they can derive an optimal partition with any type of data. The **fanny** fuzzy clustering algorithm was used in the analysis of the epilepsy data, as, contrary to other fuzzy methods, the data input can also be a dissimilarity matrix, it is more robust to the spherical cluster assumption and certain tools exist which allow the quality assessment of the results obtained by a fuzzy clustering method. Various **fanny** methods were compared, based on the distance matrix in use, the number of required clusters and the value of the membership exponent. Tools such as *Dunn's partition coefficient* and *silhouette* information were used to assess the quality of the derived partitions, and thus, to select the optimal fuzzy clustering mode for the epilepsy data. Results of the analyses showed that the optimal fuzzy clustering method is the 2-cluster fuzzy partition derived by the fuzzy method using the *SqEuclidean* metric and fuzzifier value 2. Suitable statistical tests proved that there is a relationship among the patients with respect to their *Age* and *BMI* categories, whereas fuzzy clustering was not able to discriminate the patients regarding their two epilepsy clinical characteristics and their *Gender*.

In hard clustering algorithms, each and every object belongs exclusively to one and only one cluster. In addition, the derived clusters are restricted to being compact and having spherical shape. The *k*-means hard clustering algorithm was used in the analysis of the epilepsy data, a very popular clustering approach, albeit not widely used until now in metabolic profiling applications. The optimality criterion is to minimise the within-clusters sum of squares while maximising at the same time the between-clusters sum-of-squares, using the *squared Euclidean* distance metric to measure the distance between the objects and the centroids of the clusters. Due to the nature of optimisation in this algorithm, it usually produces tighter clusters than HCA. It is also computationally faster than HCA when the number of variables is large, as happens in the case of metabonomics data. The results of the analyses showed that the 2-cluster partition is the best partition derived by a *k*-means clustering method. This was found to be the same partition as the best fuzzy clustering partition, save the differences in the fitting and the silhouette width values of the two methods. Therefore, both methods have the same discriminating ability concerning the five clinical characteristics of the patients.



Competitive learning algorithms form a different category of clustering methods than those previously mentioned. In this case, for each object presented to the algorithm, all the predefined representatives in a set compete with each other and the winner is the representative which is closer, using some distance measure, to the object. Consequently, the winner representative is being updated to be closer to the object, with the procedure continuing for all objects, until no updates can occur in any and all representatives. The Self-organising maps technique, which was used to analyse the epilepsy data, is one such algorithm. In this case, the representatives are called codebook vectors. This is a non-linear method for dimension reduction and visualisation of data (like NLM), as well as for unsupervised classification (like a clustering algorithm). That is possible, as the SOM provides a map-like visualisation of a multidimensional input space to a usually two-dimensional array of nodes, with each node associated with a codebook vector. Contrary to PCA and MDS, an important feature of SOM is its granularity, as the map consists of a discrete array and mapping to intermediate positions between the nodes is not defined. This granularity depends on the number of nodes in the map and the width of the neighbour function used for the updates in the algorithm. Two maps were used for the SOM analyses, a large one of (as recommended in the literature) size  $24 \times 2$  nodes (with respect to the number of patients in the data) and a smaller one of size  $3 \times 2$  for comparison purposes. A map of fewer than 6 nodes is not recommended, as, if a map has very few nodes, it will almost certainly fail to represent faithfully the distribution of the input data. The analyses for both maps, concentrating on the 6-cluster method defined by the 6 node map, showed that the SOM algorithm was capable of discriminating the patients according to *Gender*, *Age* and *BMI*, but not with respect to their *Seizure Type* and *Response to AEDs*. In addition, the appropriate graphical tools proved that the most important spectral area, for the differences in the intensity levels of the variables in the six clusters, is in the range 1 - 1.6 *ppm*. The four variables with the highest mean values were found to be related mainly to clusters 3 and 4, with patients 116 and 122 in cluster 3, as well as patients 1 and 15 in cluster 4, being most closely associated with those variables.

Despite all the findings in the analyses described previously, a common characteristic of all clustering algorithms used was their incapability of distinguishing the patients with regards to their epilepsy characteristics, and more importantly their *Response to AEDs*. That essentially means that further investigation is needed, to establish whether the failure of the exploratory methods is caused by the available epilepsy data or not and in what way. More specifically, the assessment of two possibilities: a) No difference exists in the spectrum between the responders and non-responders to AEDs in the data, so that no exploratory technique can illustrate it, and b) There is a difference, but the methods applied cannot illustrate it, due to either the sample size being too small or the difference being too small.

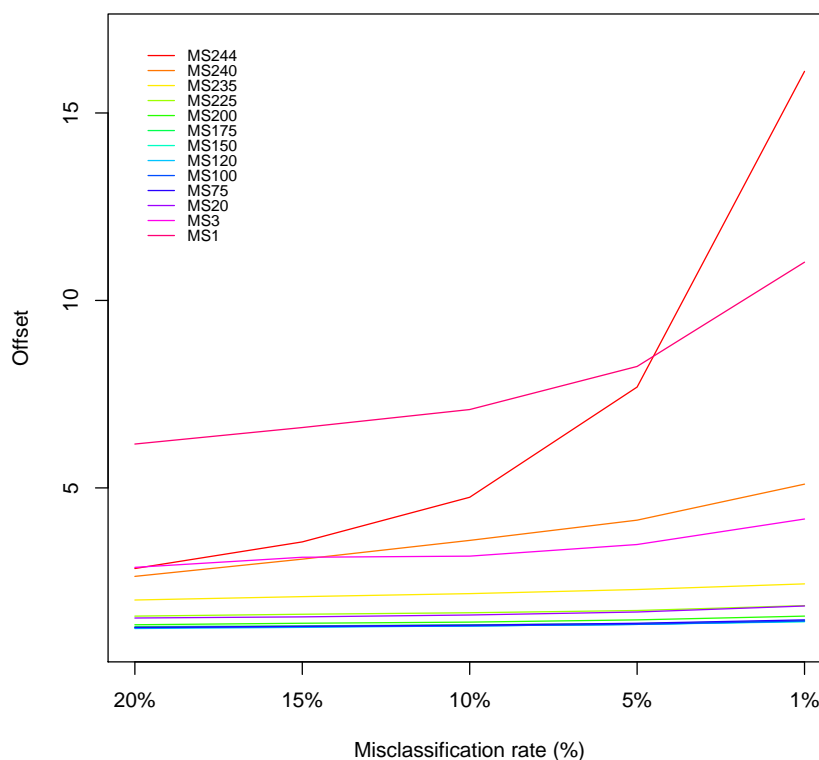
To answer which of the two possibilities is valid in this case, a series of simulation studies based on the epilepsy data was presented in Chapter 8. In each simulation case, two data sets were generated and used to investigate the capability of PCA to discriminate between two groups of points, i.e. a reference and a test data set. This involved the mean-shifting of subsets of variables in the test set, containing 244, 120, 20, 3 and 1 variables in each simulation experiment. The selected variables in each experiment were chosen in three different ways, according to decreasing and increasing order of their standard deviation values and decreasing order of their mean values. Simulation experiments were performed, with each data set having 100, 500 and 1000 samples in each experiment, to allow for conclusions on whether the sample size plays any role in the discriminating ability of PCA. This ability was assessed using two statistics, namely the *LDA misclassification rate* and the *average separation* of two distributions of points. Results showed that the discriminating ability of PCA depends on the number of mean-shifted variables. Also, the results of the simulation experiments proved to be independent of the sample size of the two data sets. Selecting the variables to mean-shift according to their increasing standard deviation results in the smallest offsets among all three variable selection methods in the experiments, with the decreasing means method of selection being the second best. Concerning the two statistics, their coefficient of variation values indicated that the *average separation* is more stable than the *misclassification rate*, especially for large samples sizes. Summary results of the simulation experiments for the *misclassification rate* for different subsets of mean-shifted variables, can be seen in Tables 9.3 and 9.4 for unscaled data and data row-scaled to a constant total, respectively. An illustration of the relation between *misclassification rates* and offsets for these experiments can be seen in Figures 9.1 and 9.2, for the unscaled and the row-scaled data, respectively.

In the case of the unscaled data, the *misclassification rate* becomes considerably smaller for large offsets (values above 5), when the number of mean-shifted variables is either very large ( $\geq 240$ ) or very small ( $< 5$ ), whereas in the case of the row-scaled data, this is valid only for very small subsets of mean-shifted variables ( $\leq 3$ ).

Overall, it was shown that under certain conditions, PCA (and consequently other data exploratory techniques) could be capable of discriminating the patients with respect to their epilepsy characteristics, and therefore, these methods could be important in metabolic profiling, should these conditions be met. As far as the implications for the design and analysis of the epilepsy data used in the simulation experiments are concerned, it is clear that there are not enough non-responder patients, and no differences were detected in a large number of variables, meaning that no major differences exist in this case between responders and non-responders.

**Table 9.3:** Summary results (offsets) for the various MS cases with S500 and using MAXMEAN (apart from MS244) with the UNSCALED and LOG-TRANSFORMED data. The results are for 100 runs of the simulation algorithm and the offsets correspond to multiplicative factors on the original scale of the data. In bold are shown the MS cases that were used in the simulation experiments.

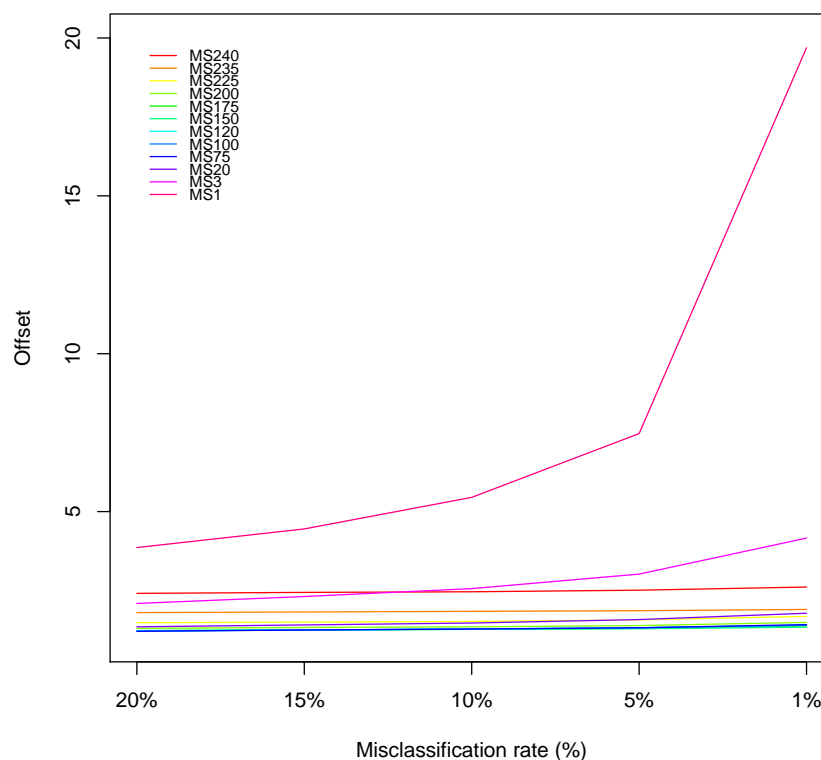
Subset of Variables	Misclassification Rate				
	20%	15%	10%	5%	1%
<b>MS244</b>	2.85	3.56	4.75	7.69	16.11
MS240	2.64	3.10	3.60	4.14	5.10
MS235	2.01	2.10	2.18	2.29	2.44
MS225	1.58	1.63	1.67	1.73	1.86
MS200	1.35	1.39	1.42	1.48	1.58
MS175	1.30	1.32	1.35	1.38	1.46
MS150	1.27	1.30	1.32	1.36	1.43
<b>MS120</b>	1.25	1.28	1.32	1.36	1.44
MS100	1.27	1.30	1.32	1.36	1.45
MS75	1.28	1.31	1.34	1.39	1.48
<b>MS20</b>	1.53	1.56	1.61	1.69	1.85
<b>MS3</b>	2.88	3.15	3.18	3.49	4.17
<b>MS1</b>	6.17	6.61	7.09	8.24	11.02



**Figure 9.1:** Misclassification rate vs offset for various MS cases with the UNSCALED and LOG-TRANSFORMED data. The experiments are the same as described in Table 9.3.

**Table 9.4:** Summary results (offsets) for the various MS cases with S500 and using MAXMEAN (apart from MS244) with the ROW-SCALED and LOG-TRANSFORMED data. The results are for 100 runs of the simulation algorithm and the offsets correspond to multiplicative factors on the original scale of the data. In bold are shown the MS cases that were used in the simulation experiments.

Subset of Variables	Misclassification Rate				
	20%	15%	10%	5%	1%
<b>MS244</b>	—	—	—	—	—
MS240	2.41	2.44	2.46	2.51	2.61
MS235	1.80	1.82	1.84	1.86	1.90
MS225	1.48	1.50	1.52	1.57	1.68
MS200	1.30	1.33	1.35	1.39	1.49
MS175	1.23	1.25	1.27	1.30	1.38
MS150	1.23	1.25	1.27	1.29	1.34
<b>MS120</b>	1.22	1.25	1.28	1.32	1.40
MS100	1.21	1.25	1.28	1.32	1.40
MS75	1.21	1.25	1.28	1.32	1.42
<b>MS20</b>	1.35	1.41	1.47	1.58	1.78
<b>MS3</b>	2.09	2.31	2.56	3.02	4.16
<b>MS1</b>	3.86	4.45	5.45	7.47	19.69



**Figure 9.2:** Misclassification rate vs offset for various MS cases with the ROW-SCALED and LOG-TRANSFORMED data. The experiments are the same as described in Table 9.4.

## 9.2 Further Recommendations

As has already been seen, classical PCA methods are based on the empirical covariance matrix of the data, which means that they can be affected by the presence of outliers. For this reason, it is often recommended that a robust PCA method should be used if there are indications of the presence of potential outliers. In Chapter 5, an investigation was presented concerning the possibility of any existing outliers affecting the results of the PCA. It was shown that when the potential outliers indicated by certain tests are removed from the data and the PCA is applied again, there are no great changes to the results, compared with those of the full data set. However, it would be prudent, if for example, a robust PCA method (such as the Grid search algorithm, proposed and described in [Croux et al. \(2007\)](#)), is used for the analysis of epilepsy data, to ensure that indeed no effects from any potential outliers are observed in the results of the analyses. A comparison of the interpretation of the results derived from *Classic* and *Grid* PCA, with and without the outlier samples in the data, can be seen in Table 9.5. In the *Grid*

**Table 9.5:** Interpretation of the derived PCs in the two PCA methods. The data for both methods are the same as those used in the analyses of Chapter 5. The quotes “ mean that results were the same as the corresponding results for the data without the outliers removed.

	Classic PCA	Grid PCA
<b>Gender</b>	PC2	PC3
<b>Seizure type</b>	PC1, PC2, PC3 (IGE) (+)	PC1, PC3 (IGE) (-)
<b>Response</b>	-	-
<b>Age</b>	PC1, PC3 [16-26] (+) PC2 [26-47] (+) [47-99] (-)	PC1 [16-26] (-) PC2 [26-47] (-) PC3 [47-99] (+)
<b>BMI</b>	PC1 [16-22] (22-25) (28-45.1) (+) PC3 (25-28] (+)	PC1 [16-22] (22-25] (-) PC2 [25-28] (28-45.1] (-)
	<b>Removed outliers - Classic</b>	<b>Removed outliers - Grid</b>
<b>Gender</b>	”	PC4
<b>Seizure type</b>	”	”
<b>Response</b>	”	”
<b>Age</b>	”	”
<b>BMI</b>	”	PC3 (25-28] (28-45.1] (-)

PCA results, the R function `PCAggrid` of package `pcaPP` was used, with the squared median absolute deviation being the robust variance estimator. The indicative results in Table 9.5 show that, for instance, regarding the *Age* categories of the patients, the *Grid* PCA results are easier to interpret, with each of the first three PCs associated with just one *Age* category, whereas in the case of the *Classic* PCA, two PCs, i.e. PC1 and PC3, are associated with the young patients, and PC2 with the patients in the rest of the *Age* categories. The same can be said for *BMI*. Further experimentation is needed to assess the usefulness of robust PCA methods for the epilepsy data.

Considering the results obtained by PCA in Chapter 5, another useful aspect of the analyses is often to assess the significance of the variables to these results. There are many statistical methods which can be used to answer this question (Brereton, 2009). For example, two of these methods are the *t-statistic* and the *Fisher weight*. The former is a univariate method used to determine which variables differ most between two groups, examining the ratio of the difference of the means and the pooled standard deviation of the intensities of each variable. In the case of more than two classes, this statistic can be applied as a series of one vs all comparisons. The *Fisher weight* assesses the significance of variables by using the ratio of within class variance to between class variance, defined by

$$f_j = \frac{\sum_{g=1}^G I_g (\bar{x}_{jg} - \bar{x}_j)^2}{s_{jpool}^2 \sum_{g=1}^G (I_g - 1)},$$

for  $G$  classes, where  $\bar{x}_j$  is the mean of variable  $j$  over all classes,  $\bar{x}_{jg}$  the mean of variable  $j$  for class  $g$ ,  $I_g$  the number of samples in class  $g$  and  $s_{jpool}$  is a weighted mean of the standard deviation  $s$  within each class (and not the overall standard deviation over all samples). Unlike the *t-statistic*, the *Fisher weight* can be computed for any number of classes.

In fuzzy clustering, a recent development is the introduction of a new algorithm, namely DiffFUZZY, which, according to its creators, yields better results than traditional fuzzy clustering algorithms in certain cases (Cominetti et al., 2010). More specifically, in cases where the data sets to be analysed are of convex shape, the results are similar to those of the traditional fuzzy methods, but when the data contains clusters with a complex, non-linear geometric structure (e.g. curved or elongated or clusters of different dispersion), only DiffFUZZY can handle them successfully. In addition, DiffFUZZY does not require any information in advance concerning the number of clusters in the data.

Another method which could result in better clustering of the data involves the combination of the fuzzy *c*-means (FCM) method with partial least squares (PLS), which, according to Li et al. (2009), can be successful in modelling the metabolic profiles to be analysed, in cases where other multivariate explorative techniques fail. This happens because the combination of FCM with PLS allows for better optimisation of the two parameters of the method, the number of clusters and the fuzziness coefficient. Examples are given, to illustrate the efficiency of this method in metabolomics, compared to other multivariate techniques such as PCA.

The SOM method, which was described in Chapter 7, was found to be quite useful for clustering metabolomics data. However, a drawback of this technique is that the SOM's structure is fixed and has to be determined in advance. This effectively means that the method is restricted by its structure in finding groups in a more natural way.

A technique which was found to overcome this restriction is the *growing self-organizing map* (GSOM). This method allows for a dynamic structure with the spread of the map controlled by a parameter called spread factor. This method has been applied successfully in biomedical data discovery, such as in class discovery from leukemia and colon cancer microarray data (Hsu et al., 2003), and in the improvement of the binning process in environmental whole-genome shotgun sequencing (Chan et al., 2008). The effect of spread factor value to the cluster formation and separation in GSOM is investigated by Ahmad et al. (2010).

A general recommendation for improving the results of the analyses in Chapters 5 to 7 is the increase of the resolution of the NMR spectra, obtaining and analysing thus metabonomics data sets with a higher number of metabolites (variables). This could be achieved during the pre-processing of the signals, by using a smaller bin size of, e.g. 0.01 ppm, instead of that used in the analyses in the thesis (0.04 ppm).

Regarding the simulation experiments, two things could be considered for further experimentation, i.e. the sample size of the generated data sets and the offsets used to mean-shift the selected variables. In the experiments of Chapter 8, both the reference set and the test set in each experiment had the same sample size. Another possibility is to investigate what will happen when the two data sets have different sample sizes, with either the reference set having a larger sample size than the test set or vice versa. Concerning the offsets, the following questions could be investigated:

- In all the experiments done, the offsets were added to the variables. What would be the outcome of the experiments in the case of adding the offset to some of the variables and subtracting the same offset from other variables?
- In addition, so far all offsets were the same for all variables. Would there have been any difference in the outcome of the experiments, if offsets of different sizes were used?
- Would there have been any difference in the case of a random choice of offsets ?

Finally, concerning the method to select the variables to mean-shift, if the variables were chosen according to decreasing order of their correlation, would the results of the experiments been affected in any way?

The clustering methods described in Chapter 7 and applied in the epilepsy data, were used to explore the data structure for potential groups and the features which distinguish these groups from each other. Another important step in chemometrics and the analysis of metabonomics data is the application of supervised techniques (described in Section 8.2) to the grouping information of the samples obtained from the unsupervised clustering methods or groups known from previously recorded data. There are two types of *classifiers*, that is, discrimination rules for classification purposes (Breteon, 2009):

1. Two- or more (multi) class *classifiers*, in which the samples in a data set are

assigned to one of two (or more, in the case of multi-class *classifiers*) groups, e.g. *linear* (LDA), *partial least squares* (PLS-DA) (Lindon et al., 2001) and *quadratic discriminant analysis* (QDA) as well as *learning vector quantization* (LVQ) and *support vector machines* (SVMs) (Brereton, 2009).

2. One-class *classifiers*, in which each group is modelled separately, i.e. *soft independent modelling of class analogy* (SIMCA) (Lindon et al., 2001; Brereton, 2009) and *support vector data description* (SVDD) (Brereton, 2009).

Then, for example, the selected *classifier* could be used to predict the class of the remaining (25 patients with unclassified *Response to AEDs*) in the original epilepsy data set of the 122 patients, classifying them to either the responders or non-responders. Such a *classifier* could predict the class of the patients of any other epilepsy data set with samples of similar type to those in the epilepsy data set.



# APPENDICES

---

## List of Appendices

### Contents

1. **Appendix A** - Lists of Mean-Shifted Variables
2. **Appendix B** - Simulation Results
3. **Appendix C** - R Code Used in the Simulation Algorithm
4. **Appendix D** - Vignette for the Simulation Algorithm
5. **Appendix E** - Lists of components of Mass Spectrometers

# Appendix A

---

## Lists of Mean-Shifted Variables

### Contents

1. Mean-shifted variables using *MAXDEV*
2. Mean-shifted variables using *MINDEV*
3. Mean-Shifted variables using *MAXMEAN*

**Table A.1:** Mean-Shifted Variables for the MAXDEV cases. The numbers in normal typeface are the standard deviations of the 244 original variables for the 97 patients with epilepsy.

<b>MS120</b>									
<b>5.78</b>	<b>5.82</b>	<b>5.98</b>	<b>6.02</b>	<b>6.30</b>	<b>5.74</b>	<b>4.66</b>	<b>6.26</b>	<b>4.62</b>	<b>1.30</b>
0.876	0.787	0.681	0.671	0.658	0.658	0.657	0.656	0.656	0.654
<b>5.94</b>	<b>5.34</b>	<b>5.86</b>	<b>6.06</b>	<b>2.46</b>	<b>7.74</b>	<b>7.78</b>	<b>3.34</b>	<b>6.22</b>	<b>7.06</b>
0.652	0.650	0.649	0.648	0.647	0.647	0.646	0.646	0.642	0.642
<b>4.58</b>	<b>7.82</b>	<b>8.86</b>	<b>8.90</b>	<b>8.82</b>	<b>6.18</b>	<b>7.54</b>	<b>6.78</b>	<b>7.58</b>	<b>8.94</b>
0.641	0.641	0.640	0.640	0.640	0.640	0.639	0.639	0.639	0.639
<b>7.86</b>	<b>7.70</b>	<b>6.34</b>	<b>6.82</b>	<b>2.62</b>	<b>7.62</b>	<b>6.58</b>	<b>6.14</b>	<b>8.98</b>	<b>9.02</b>
0.638	0.638	0.638	0.638	0.638	0.638	0.637	0.637	0.637	0.637
<b>7.02</b>	<b>8.78</b>	<b>7.46</b>	<b>8.54</b>	<b>9.06</b>	<b>6.38</b>	<b>2.50</b>	<b>8.50</b>	<b>8.14</b>	<b>9.10</b>
0.637	0.637	0.637	0.637	0.637	0.637	0.637	0.637	0.636	0.636
<b>8.58</b>	<b>9.14</b>	<b>2.58</b>	<b>5.30</b>	<b>7.38</b>	<b>6.62</b>	<b>3.10</b>	<b>9.18</b>	<b>6.46</b>	<b>7.90</b>
0.636	0.636	0.636	0.636	0.636	0.636	0.636	0.636	0.636	0.636
<b>8.46</b>	<b>6.10</b>	<b>8.42</b>	<b>8.10</b>	<b>4.42</b>	<b>6.42</b>	<b>8.18</b>	<b>8.74</b>	<b>2.42</b>	<b>4.38</b>
0.636	0.636	0.635	0.635	0.635	0.635	0.635	0.635	0.635	0.635
<b>3.38</b>	<b>6.90</b>	<b>4.14</b>	<b>8.62</b>	<b>8.22</b>	<b>9.58</b>	<b>4.50</b>	<b>7.50</b>	<b>5.26</b>	<b>7.42</b>
0.634	0.634	0.634	0.634	0.634	0.634	0.634	0.634	0.634	0.634
<b>4.54</b>	<b>7.94</b>	<b>8.02</b>	<b>9.22</b>	<b>9.26</b>	<b>8.38</b>	<b>7.98</b>	<b>8.70</b>	<b>8.34</b>	<b>9.54</b>
0.634	0.634	0.634	0.634	0.633	0.633	0.633	0.633	0.633	0.633
<b>7.10</b>	<b>8.66</b>	<b>9.50</b>	<b>6.54</b>	<b>8.26</b>	<b>6.98</b>	<b>8.30</b>	<b>6.74</b>	<b>4.46</b>	<b>9.62</b>
0.633	0.633	0.633	0.633	0.633	0.633	0.633	0.632	0.632	0.632
<b>9.30</b>	<b>8.06</b>	<b>6.86</b>	<b>3.06</b>	<b>4.22</b>	<b>6.66</b>	<b>2.66</b>	<b>6.70</b>	<b>9.66</b>	<b>6.50</b>
0.632	0.632	0.632	0.632	0.632	0.632	0.632	0.632	0.632	0.631
<b>7.14</b>	<b>2.70</b>	<b>7.30</b>	<b>2.10</b>	<b>9.98</b>	<b>7.34</b>	<b>9.34</b>	<b>3.30</b>	<b>4.18</b>	<b>3.22</b>
0.631	0.631	0.631	0.631	0.631	0.631	0.630	0.631	0.630	0.630
<b>MS20</b>									
<b>5.78</b>	<b>5.82</b>	<b>5.98</b>	<b>6.02</b>	<b>6.30</b>	<b>5.74</b>	<b>4.66</b>	<b>6.26</b>	<b>4.62</b>	<b>1.30</b>
0.876	0.787	0.681	0.671	0.658	0.658	0.657	0.656	0.656	0.654
<b>5.94</b>	<b>5.34</b>	<b>5.86</b>	<b>6.06</b>	<b>2.46</b>	<b>7.74</b>	<b>7.78</b>	<b>3.34</b>	<b>6.22</b>	<b>7.06</b>
0.652	0.650	0.649	0.648	0.647	0.647	0.646	0.646	0.642	0.642
<b>MS3</b>									
<b>5.78</b>	<b>5.82</b>	<b>5.98</b>							
0.876	0.787	0.681							
<b>MS1</b>									
<b>5.78</b>									
0.876									

**Table A.2:** Mean-Shifted Variables for the MINDEV cases. The numbers in normal typeface are the standard deviations of the 244 original variables for the 97 patients with epilepsy.

<b>MS120</b>									
<b>4.98</b>	<b>4.94</b>	<b>5.02</b>	<b>5.06</b>	<b>5.10</b>	<b>0.90</b>	<b>5.14</b>	<b>5.18</b>	<b>5.54</b>	<b>0.70</b>
0.584	0.585	0.592	0.599	0.604	0.605	0.606	0.607	0.609	0.609
<b>0.86</b>	<b>5.42</b>	<b>0.14</b>	<b>5.46</b>	<b>5.22</b>	<b>2.02</b>	<b>1.18</b>	<b>5.38</b>	<b>0.62</b>	<b>1.58</b>
0.609	0.609	0.611	0.611	0.611	0.611	0.611	0.612	0.612	0.612
<b>0.02</b>	<b>1.38</b>	<b>0.74</b>	<b>2.06</b>	<b>0.58</b>	<b>0.66</b>	<b>5.50</b>	<b>3.66</b>	<b>0.18</b>	<b>0.94</b>
0.612	0.612	0.612	0.613	0.613	0.613	0.613	0.613	0.613	0.614
<b>0.34</b>	<b>1.02</b>	<b>0.30</b>	<b>0.78</b>	<b>0.38</b>	<b>2.26</b>	<b>1.14</b>	<b>0.26</b>	<b>0.42</b>	<b>1.26</b>
0.614	0.614	0.614	0.614	0.614	0.614	0.615	0.615	0.615	0.615
<b>5.58</b>	<b>0.22</b>	<b>0.54</b>	<b>0.10</b>	<b>4.30</b>	<b>5.90</b>	<b>0.06</b>	<b>1.10</b>	<b>1.42</b>	<b>1.22</b>
0.615	0.615	0.616	0.616	0.616	0.616	0.616	0.617	0.617	0.617
<b>1.62</b>	<b>0.98</b>	<b>0.50</b>	<b>0.82</b>	<b>2.22</b>	<b>0.46</b>	<b>1.54</b>	<b>1.46</b>	<b>3.62</b>	<b>3.86</b>
0.617	0.617	0.617	0.617	0.617	0.618	0.618	0.618	0.620	0.620
<b>1.94</b>	<b>4.26</b>	<b>3.90</b>	<b>1.82</b>	<b>1.98</b>	<b>3.70</b>	<b>2.18</b>	<b>1.06</b>	<b>1.70</b>	<b>2.94</b>
0.620	0.620	0.620	0.620	0.635	0.620	0.621	0.621	0.621	0.622
<b>1.66</b>	<b>3.26</b>	<b>2.86</b>	<b>4.06</b>	<b>3.58</b>	<b>2.74</b>	<b>1.50</b>	<b>1.86</b>	<b>2.98</b>	<b>5.62</b>
0.622	0.623	0.623	0.623	0.624	0.624	0.624	0.624	0.624	0.625
<b>7.22</b>	<b>2.82</b>	<b>2.34</b>	<b>4.10</b>	<b>3.82</b>	<b>3.46</b>	<b>2.78</b>	<b>3.78</b>	<b>3.42</b>	<b>2.30</b>
0.625	0.625	0.625	0.625	0.625	0.625	0.625	0.625	0.625	0.626
<b>3.98</b>	<b>7.26</b>	<b>7.66</b>	<b>2.38</b>	<b>5.66</b>	<b>3.18</b>	<b>7.18</b>	<b>3.74</b>	<b>5.70</b>	<b>4.02</b>
0.626	0.626	0.626	0.626	0.626	0.626	0.627	0.627	0.627	0.627
<b>1.90</b>	<b>6.94</b>	<b>3.54</b>	<b>1.74</b>	<b>2.54</b>	<b>2.14</b>	<b>9.42</b>	<b>4.34</b>	<b>3.02</b>	<b>9.94</b>
0.627	0.627	0.628	0.628	0.628	0.628	0.628	0.628	0.628	0.628
<b>1.78</b>	<b>9.38</b>	<b>9.74</b>	<b>9.78</b>	<b>1.34</b>	<b>9.82</b>	<b>9.90</b>	<b>9.46</b>	<b>3.50</b>	<b>9.86</b>
0.628	0.628	0.628	0.629	0.629	0.629	0.629	0.629	0.629	0.629
<b>MS20</b>									
<b>4.98</b>	<b>4.94</b>	<b>5.02</b>	<b>5.06</b>	<b>5.10</b>	<b>0.90</b>	<b>5.14</b>	<b>5.18</b>	<b>5.54</b>	<b>0.70</b>
0.584	0.585	0.592	0.599	0.604	0.605	0.606	0.607	0.609	0.609
<b>0.86</b>	<b>5.42</b>	<b>0.14</b>	<b>5.46</b>	<b>5.22</b>	<b>2.02</b>	<b>1.18</b>	<b>5.38</b>	<b>0.62</b>	<b>1.58</b>
0.609	0.609	0.611	0.611	0.611	0.611	0.611	0.612	0.612	0.612
<b>MS3</b>									
<b>4.98</b>	<b>4.94</b>	<b>5.02</b>							
0.584	0.585	0.592							
<b>MS1</b>									
<b>4.98</b>									
0.584									

**Table A.3:** Mean-Shifted Variables for the MAXMEAN cases. The numbers in normal typeface are the means of the 244 original variables for the 97 patients with epilepsy.

<b>MS120</b>									
<b>1.30</b>	<b>1.26</b>	<b>0.86</b>	<b>1.22</b>	<b>0.90</b>	<b>1.34</b>	<b>2.02</b>	<b>0.82</b>	<b>0.94</b>	<b>1.18</b>
24.05	24.04	23.70	23.62	23.54	23.52	23.30	23.29	23.26	23.16
<b>0.98</b>	<b>2.06</b>	<b>1.98</b>	<b>1.58</b>	<b>1.50</b>	<b>1.54</b>	<b>1.02</b>	<b>1.14</b>	<b>1.46</b>	<b>1.38</b>
23.14	23.12	23.04	23.02	23.01	23.00	22.99	22.95	22.95	22.93
<b>1.06</b>	<b>1.10</b>	<b>1.42</b>	<b>0.78</b>	<b>1.70</b>	<b>1.62</b>	<b>3.22</b>	<b>1.94</b>	<b>1.74</b>	<b>2.10</b>
22.92	22.89	22.87	22.80	22.80	22.78	22.78	22.77	22.77	22.75
<b>1.66</b>	<b>2.22</b>	<b>2.26</b>	<b>1.78</b>	<b>1.90</b>	<b>1.82</b>	<b>1.86</b>	<b>3.90</b>	<b>0.74</b>	<b>2.14</b>
22.74	22.73	22.71	22.68	22.67	22.67	22.66	22.65	22.61	22.59
<b>2.30</b>	<b>2.18</b>	<b>0.70</b>	<b>2.34</b>	<b>3.74</b>	<b>3.70</b>	<b>0.66</b>	<b>3.82</b>	<b>3.26</b>	<b>3.86</b>
22.56	22.51	22.51	22.50	22.41	22.40	22.40	22.39	22.37	22.33
<b>4.10</b>	<b>2.38</b>	<b>4.06</b>	<b>3.66</b>	<b>3.98</b>	<b>2.42</b>	<b>3.02</b>	<b>3.94</b>	<b>4.14</b>	<b>3.42</b>
22.33	22.31	22.31	22.31	22.30	22.27	22.27	22.27	22.27	22.26
<b>3.78</b>	<b>5.30</b>	<b>4.02</b>	<b>2.46</b>	<b>0.62</b>	<b>4.26</b>	<b>3.46</b>	<b>4.30</b>	<b>2.98</b>	<b>4.18</b>
22.26	22.24	22.23	22.23	22.21	22.17	22.16	22.15	22.15	22.14
<b>3.54</b>	<b>2.94</b>	<b>2.74</b>	<b>4.22</b>	<b>2.50</b>	<b>0.58</b>	<b>3.50</b>	<b>3.06</b>	<b>2.70</b>	<b>2.90</b>
22.10	22.07	22.07	22.07	22.06	22.05	22.04	22.00	22.00	21.97
<b>3.58</b>	<b>2.78</b>	<b>2.54</b>	<b>0.54</b>	<b>4.34</b>	<b>3.62</b>	<b>5.26</b>	<b>4.42</b>	<b>4.46</b>	<b>3.18</b>
21.95	21.95	21.94	21.92	21.92	21.92	21.90	21.89	21.89	21.88
<b>3.10</b>	<b>4.38</b>	<b>2.66</b>	<b>4.50</b>	<b>2.58</b>	<b>0.50</b>	<b>2.86</b>	<b>2.82</b>	<b>3.34</b>	<b>2.62</b>
21.86	21.84	21.81	21.80	21.80	21.80	21.79	21.75	21.74	21.73
<b>3.14</b>	<b>3.38</b>	<b>4.66</b>	<b>3.30</b>	<b>0.46</b>	<b>4.54</b>	<b>7.06</b>	<b>0.42</b>	<b>7.10</b>	<b>7.02</b>
21.72	21.72	21.70	21.69	21.68	21.68	21.63	21.58	21.58	21.54
<b>4.58</b>	<b>7.18</b>	<b>7.14</b>	<b>7.22</b>	<b>6.98</b>	<b>0.38</b>	<b>5.34</b>	<b>7.26</b>	<b>7.30</b>	<b>5.22</b>
21.53	21.53	21.52	21.51	21.51	21.49	21.49	21.48	21.48	21.42
<b>MS20</b>									
<b>1.30</b>	<b>1.26</b>	<b>0.86</b>	<b>1.22</b>	<b>0.90</b>	<b>1.34</b>	<b>2.02</b>	<b>0.82</b>	<b>0.94</b>	<b>1.18</b>
24.05	24.04	23.70	23.62	23.54	23.52	23.30	23.29	23.26	23.16
<b>0.98</b>	<b>2.06</b>	<b>1.98</b>	<b>1.58</b>	<b>1.50</b>	<b>1.54</b>	<b>1.02</b>	<b>1.14</b>	<b>1.46</b>	<b>1.38</b>
23.14	23.12	23.04	23.02	23.01	23.00	22.99	22.95	22.95	22.93
<b>MS3</b>									
<b>1.30</b>	<b>1.26</b>	<b>0.86</b>							
24.05	24.04	23.70							
<b>MS1</b>									
<b>1.30</b>									
24.05									

# Appendix B

---

## Simulation Results

### Contents

1. Appendix B.1 - *MINDEV* Results
2. Appendix B.2 - *MAXMEAN* Results

# Appendix B.1

---

## MINDEV Results

### Contents

1. MINDEV - Case MS120 statistics results
2. MINDEV - Case MS20 statistics results
3. MINDEV - Case MS3 statistics results
4. MINDEV - Case MS1 statistics results
5. MINDEV - LDA Boundaries
6. MINDEV - Error Plots for case MS120
7. MINDEV - Error Plots for case MS20
8. MINDEV - Error Plots for case MS3
9. MINDEV - Error Plots for case MS1
10. MINDEV - Case MS120 CV results
11. MINDEV - Case MS20 CV results
12. MINDEV - Case MS3 CV results
13. MINDEV - Case MS1 CV results

## B.1.1 MINDEV - Case MS120 statistics results

**Table B.1:** Average LDA misclassification rates and average separation values for the case MS120, applying the MINDEV method in 100 runs of the experiment.

<b>S100</b>						
<b>Offset</b>	1.25	1.27	1.30	1.32	1.35	1.38
<b>Error Rate (%)</b>	23.45	18.19	13.63	9.43	6.34	3.94
<b>Average Separation</b>	11.29	11.35	11.50	11.52	11.53	11.77
<b>Offset</b>	1.40	1.43	1.46	1.49	1.52	1.55
<b>Error Rate (%)</b>	2.49	1.25	0.59	0.34	0.13	0.06
<b>Average Separation</b>	11.69	11.82	11.95	12.05	12.13	12.23
<b>S500</b>						
<b>Offset</b>	1.25	1.27	1.30	1.32	1.35	1.38
<b>Error Rate (%)</b>	23.09	18.65	13.29	9.39	6.19	4.01
<b>Average Separation</b>	11.32	11.40	11.47	11.51	11.61	11.71
<b>Offset</b>	1.40	1.43	1.46	1.49	1.52	1.55
<b>Error Rate (%)</b>	2.32	1.25	0.71	0.35	0.20	0.10
<b>Average Separation</b>	11.80	11.84	11.95	12.02	12.07	12.17
<b>S1000</b>						
<b>Offset</b>	1.25	1.27	1.30	1.32	1.35	1.38
<b>Error Rate (%)</b>	23.38	18.55	13.60	9.31	6.34	3.95
<b>Average Separation</b>	11.34	11.40	11.46	11.50	11.60	11.67
<b>Offset</b>	1.40	1.43	1.46	1.49	1.52	1.55
<b>Error Rate (%)</b>	2.35	1.32	0.71	0.35	0.17	0.09
<b>Average Separation</b>	11.74	11.86	11.90	12.05	12.11	12.18

Table B.1 gives the results of the MS120 experiments in the cases S100, S500 and S1000 for offsets in the range 1.25 – 1.55. It can be seen that offsets in the range 1.25 – 1.55 are required in all sample size cases, to achieve misclassification rates of  $\approx 24 - 0.1\%$  respectively. The average separation between the two data sets is  $\approx 11.3$  for a 20% misclassification rate and the two data sets are almost linearly separable when the average separation is  $\approx 12.2$  with misclassification rate less than 1%.



## B.1.2 MINDEV - Case MS20 statistics results

**Table B.2:** Average LDA misclassification rates and average separation values for the case MS20, applying the MINDEV method in 100 runs of the experiment.

<b>S100</b>						
<b>Offset</b>	1.40	1.46	1.52	1.58	1.65	1.72
<b>Error Rate (%)</b>	34.50	28.96	21.71	14.14	8.62	5.22
<b>Average Separation</b>	11.18	11.11	11.19	11.33	11.40	11.43
<b>Offset</b>	1.79	1.86	1.93	2.01	2.10	2.18
<b>Error Rate (%)</b>	2.34	1.09	0.42	0.12	0.04	0.03
<b>Average Separation</b>	11.44	11.56	11.69	11.75	11.71	11.90
<b>S500</b>						
<b>Offset</b>	1.46	1.52	1.58	1.65	1.72	1.79
<b>Error Rate (%)</b>	28.27	22.10	14.91	8.36	4.79	2.44
<b>Average Separation</b>	11.23	11.26	11.28	11.38	11.46	11.53
<b>Offset</b>	1.86	1.93	2.01	2.10	2.18	2.27
<b>Error Rate (%)</b>	1.18	0.52	0.22	0.07	0.03	0.00
<b>Average Separation</b>	11.57	11.67	11.71	11.80	11.83	11.99
<b>S1000</b>						
<b>Offset</b>	1.48	1.52	1.57	1.62	1.67	1.72
<b>Error Rate (%)</b>	26.88	21.69	16.18	11.30	7.58	4.79
<b>Average Separation</b>	11.26	11.27	11.32	11.32	11.38	11.47
<b>Offset</b>	1.77	1.82	1.88	1.93	1.99	2.05
<b>Error Rate (%)</b>	2.92	1.71	0.98	0.48	0.27	0.13
<b>Average Separation</b>	11.51	11.54	11.59	11.66	11.70	11.76

The results of the MS20 experiments in all three sample size cases can be seen in Table B.2. It can be seen that offsets in the range 1.52 – 2.05 are required in all sample size cases, to achieve misclassification rates of  $\approx 22 - 0.1\%$  respectively. The average separation between the two data sets is  $\approx 11.2$  for a 20% misclassification rate and the two data sets are almost linearly separable when the average separation is  $\approx 11.6$  with misclassification rate less than 1%.

### B.1.3 MINDEV - Case MS3 statistics results

**Table B.3:** Average LDA misclassification rates and average separation values for the case MS1, applying the MINDEV method in 100 runs of the experiment.

<b>S100</b>						
<b>Offset</b>	2.72	2.86	3.00	3.16	3.32	3.49
<b>Error Rate (%)</b>	31.51	25.21	20.54	14.59	9.94	5.54
<b>Average Separation</b>	11.16	11.26	11.24	11.28	11.27	11.34
<b>Offset</b>	3.67	3.86	4.06	4.26	4.48	4.71
<b>Error Rate (%)</b>	3.33	1.78	1.23	0.41	0.22	0.09
<b>Average Separation</b>	11.42	11.42	11.44	11.51	11.55	11.56
<b>S500</b>						
<b>Offset</b>	2.86	3.00	3.16	3.32	3.49	3.67
<b>Error Rate (%)</b>	25.87	21.15	13.69	8.55	4.83	2.71
<b>Average Separation</b>	11.23	11.21	11.34	11.31	11.38	11.37
<b>Offset</b>	3.86	4.06	4.26	4.48	4.71	4.95
<b>Error Rate (%)</b>	1.39	0.78	0.40	0.20	0.08	0.03
<b>Average Separation</b>	11.43	11.50	11.49	11.56	11.55	11.58
<b>S1000</b>						
<b>Offset</b>	2.86	3.00	3.16	3.32	3.49	3.67
<b>Error Rate (%)</b>	26.12	19.87	14.05	8.44	4.90	2.64
<b>Average Separation</b>	11.22	11.23	11.26	11.33	11.37	11.40
<b>Offset</b>	3.86	4.06	4.26	4.48	4.71	4.95
<b>Error Rate (%)</b>	1.42	0.66	0.33	0.15	0.07	0.03
<b>Average Separation</b>	11.43	11.48	11.51	11.55	11.59	11.60

Table B.3 shows the misclassification rates and average separation values of the experiments in the cases S100, S500 and S1000 for offsets in the ranges 2.72–4.71, 2.86 – 4.95 and 2.86 – 4.95 respectively. Offsets in the range  $\approx 2.86 - 4.95$  are required in all sample size cases, to achieve misclassification rates of  $\approx 25-0.05\%$  respectively. The average separation between the two data sets is  $\approx 11.2$  for a 20% misclassification rate and the two data sets are almost linearly separable when the average separation is  $\approx 11.5$  with misclassification rate less than 1%.

## B.1.4 MINDEV - Case MS1 statistics results

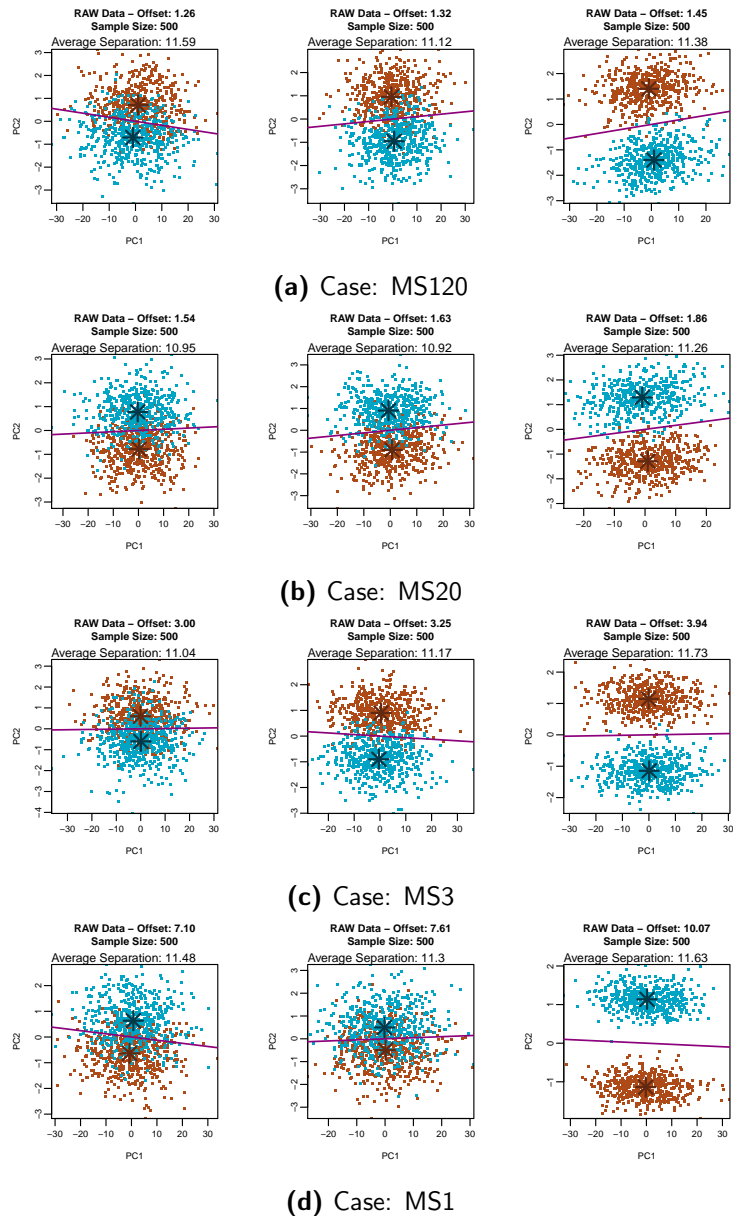
**Table B.4:** Average LDA misclassification rates and average separation values for the case MS1, applying the MINDEV method in 100 runs of the experiment.

<b>S100</b>						
<b>Offset</b>	7.03	7.39	7.77	8.17	8.58	9.03
<b>Error Rate (%)</b>	20.32	14.23	11.66	6.02	3.34	2.44
<b>Average Separation</b>	11.19	11.26	11.35	11.31	11.27	11.46
<b>Offset</b>	9.49	9.97	10.49	11.02	11.59	12.18
<b>Error Rate (%)</b>	1.98	0.97	0.25	0.55	0.13	0.06
<b>Average Separation</b>	11.34	11.41	11.46	11.47	11.48	11.53
<b>S500</b>						
<b>Offset</b>	7.03	7.39	7.77	8.17	8.58	9.03
<b>Error Rate (%)</b>	21.87	13.91	9.67	5.03	2.46	1.21
<b>Average Separation</b>	11.24	11.30	11.28	11.31	11.35	11.35
<b>Offset</b>	9.49	9.97	10.49	11.02	11.59	12.18
<b>Error Rate (%)</b>	0.54	0.19	0.14	0.08	0.02	0.00
<b>Average Separation</b>	11.45	11.40	11.45	11.43	11.47	11.54
<b>S1000</b>						
<b>Offset</b>	7.03	7.39	7.77	8.17	8.58	9.03
<b>Error Rate (%)</b>	21.05	14.29	8.63	4.34	2.16	1.04
<b>Average Separation</b>	11.26	11.28	11.34	11.32	11.34	11.37
<b>Offset</b>	9.49	9.97	10.49	11.02	11.59	12.18
<b>Error Rate (%)</b>	0.39	0.20	0.07	0.03	0.01	0.00
<b>Average Separation</b>	11.40	11.41	11.44	11.47	11.46	11.52

The results of the experiments in the cases S100, S500 and S1000 respectively, for offsets in the range 7.03 – 12.18 can be seen in Table B.4. It can be seen that offsets in the range 7.1 – 12.18 are required in all sample size cases, to achieve misclassification rates of  $\approx 20 - 0\%$  respectively. The average separation between the two data sets is  $\approx 11.2$  for a 20% misclassification rate and the two data sets are almost linearly separable when the average separation is  $\approx 11.4$  with misclassification rate less than 1%.

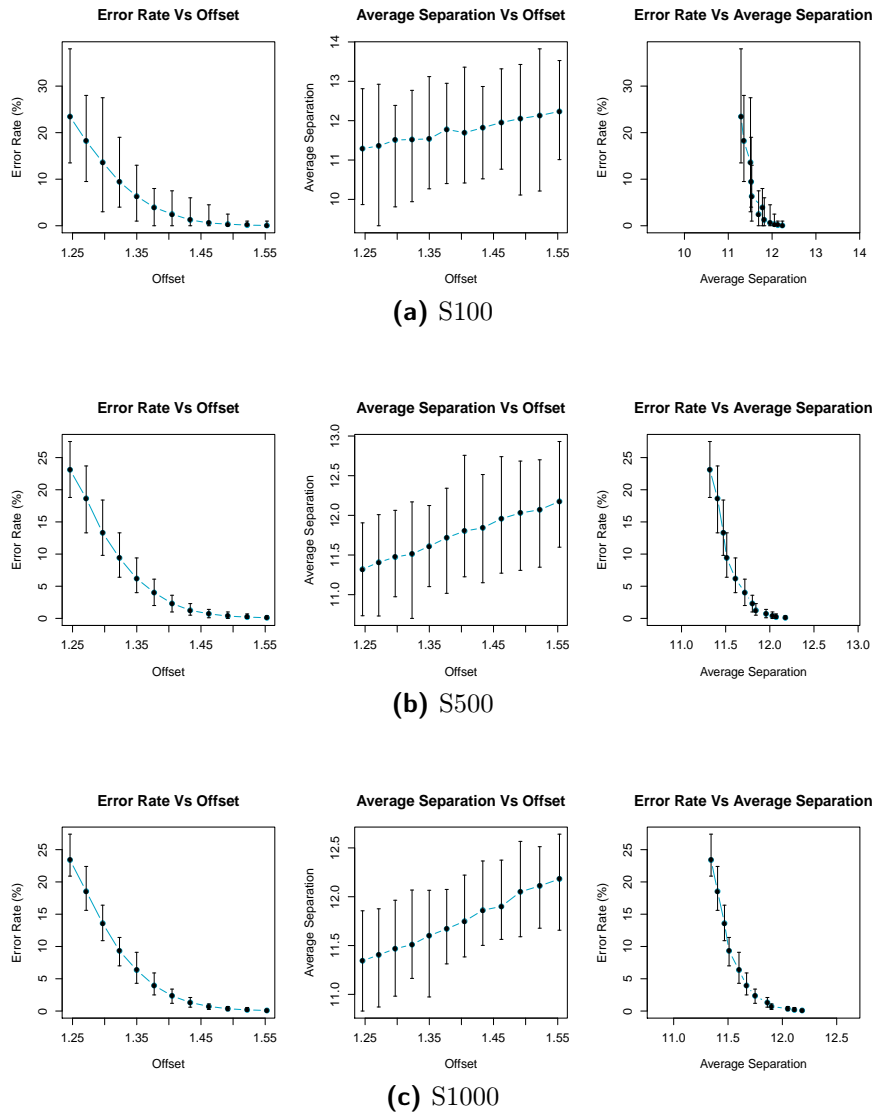
## B.1.5 MINDEV - LDA Boundaries

An illustration of how mean-shifting affects the capability of PCA to discriminate the two data sets in all four MS cases with S500 for MINDEV, superimposed with the LDA boundary for the two artificial data sets, can be seen in Figure B.1, for suitably selected offsets which correspond to 20%, 10% and 1% misclassification rates respectively.



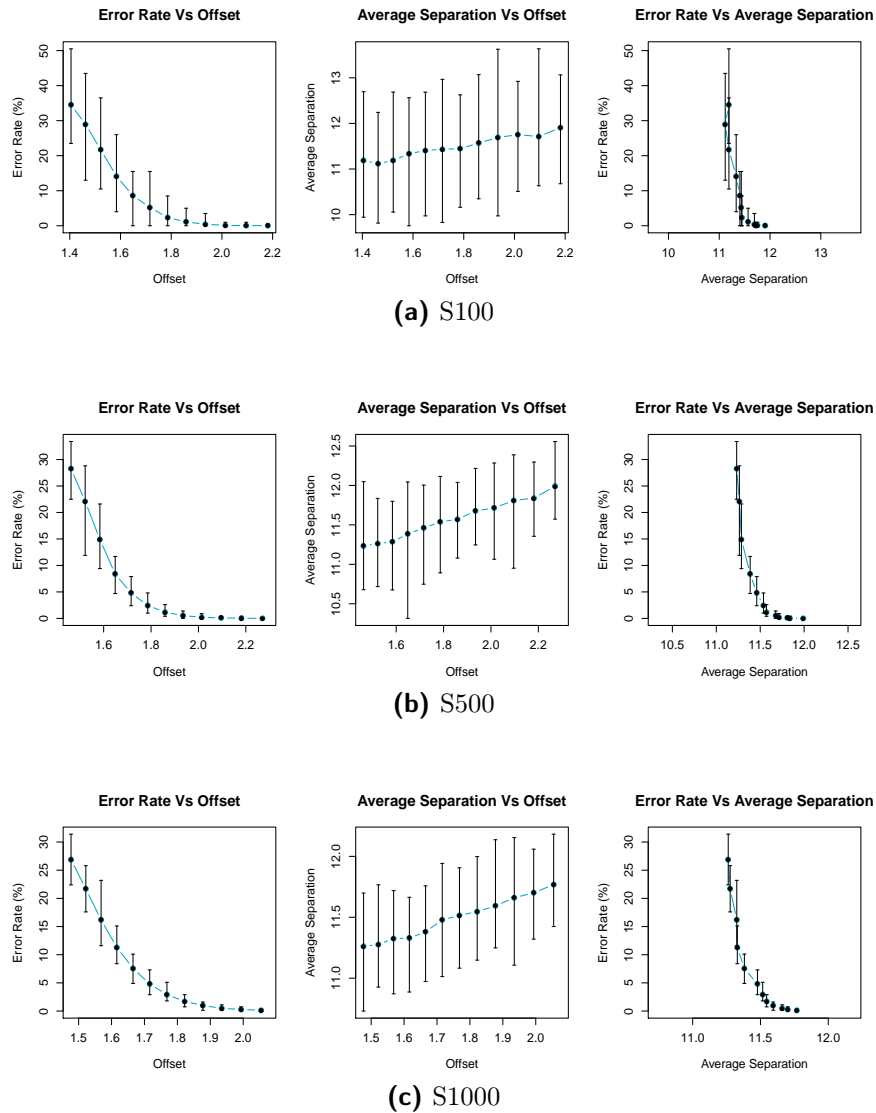
**Figure B.1:** Visualisation of the LDA boundaries for the two artificial data sets in all four cases MS (MINDEV). The data corresponds to the first two PCs for LDA. The reference and test data points are depicted in brown and blue respectively.

## B.1.6 MINDEV - Error Plots for case MS120



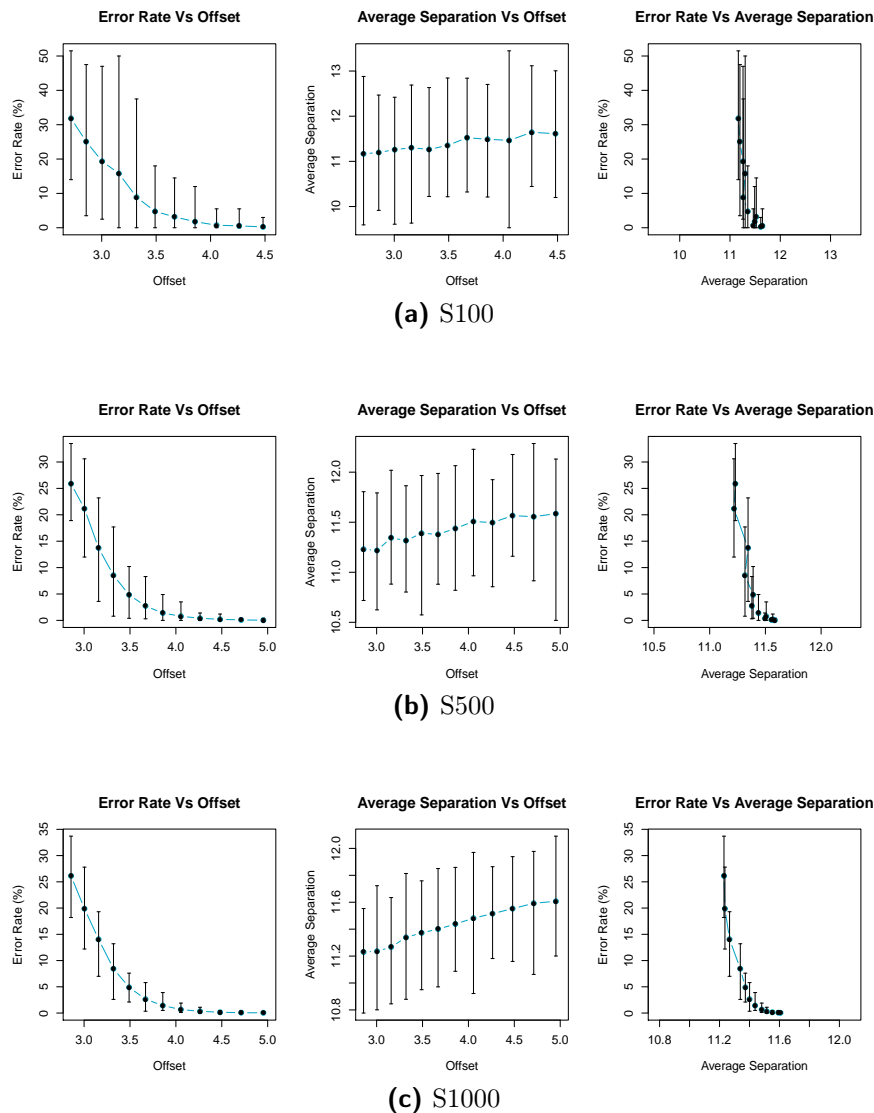
**Figure B.2:** Graphical representation of the relation among *LDA misclassification rates*, *average separation* and offsets in the case MS120 for method MINDEV. The blue lines represent the mean values of each statistic for each offset in the selected offset range. The offsets are the multiplicative factors on the original scale of the data. The vertical error bars are such that the top and bottom of a bar correspond to the maximum and minimum statistic value at the respective offset. The statistics values are the average values in 100 runs of the experiment. The two average separation plots in each sample size case are drawn using the same range of values for the *average separation*.

## B.1.7 MINDEV - Error Plots for case MS20



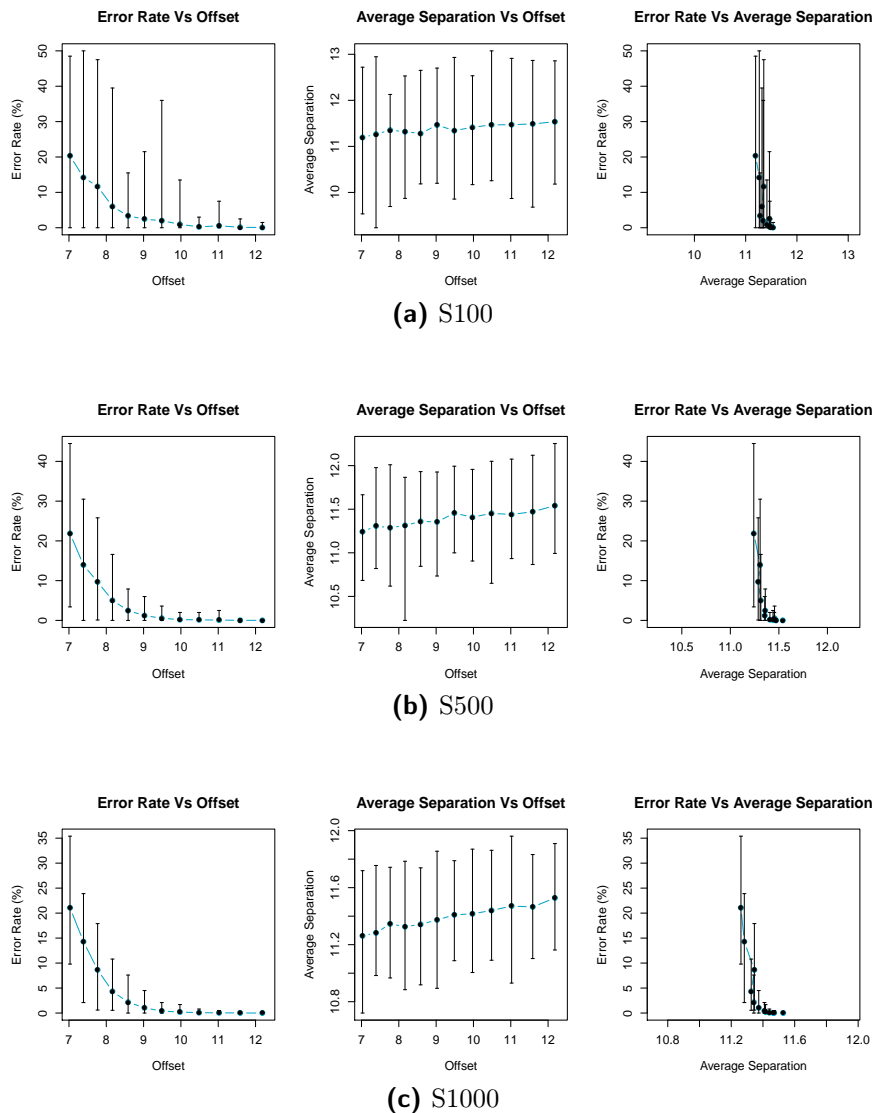
**Figure B.3:** Visualisation of the relation among *LDA misclassification rates*, *average separation* and *offsets* in the case MS20 with MINDEV. The blue lines represent the mean values of each statistic for each offset in the selected offset range. The offsets are the multiplicative factors on the original scale of the data. The vertical error bars are such that the top and bottom of a bar correspond to the maximum and minimum statistic value at the respective offset. The statistics values are the average values in 100 runs of the experiment. The two *average separation* plots in each sample size case are drawn using the same range of values for the *average separation*.

## B.1.8 MINDEV - Error Plots for case MS3



**Figure B.4:** Visualisation of the relation among *LDA misclassification rates*, *average separation* and *offsets* in the case MS3 with MINDEV. The blue lines represent the mean values of each statistic for each offset in the selected offset range. The offsets are the multiplicative range. The vertical error bars are such that the top and bottom of a bar correspond to the maximum and minimum statistic value at the respective offset. The statistics values are the average values in 100 runs of the experiment. The two average separation plots in each sample size case are drawn using the same range of values for the *average separation*.

## B.1.9 MINDEV - Error Plots for case MS1



**Figure B.5:** Visualisation of the relation among *LDA misclassification rates*, *average separation* and *offsets* in the case MS1 with MINDEV. The blue lines represent the mean values of each statistic for each offset in the selected offset range. The offsets are the multiplicative factors on the original scale of the data. The vertical error bars are such that the top and bottom of a bar correspond to the maximum and minimum statistic value at the respective offset. The statistics values are the average values in 100 runs of the experiment. The two average separation plots in each sample size case are drawn using the same range of values for the *average separation*.



## B.1.10 MINDEV - Case MS120 CV results

**Table B.5:** Coefficient of variation results for case MS120 using method MINDEV, of the *LDA* misclassification rates and average separation values in 100 runs of the experiment.

S100						
Offset	1.25	1.27	1.30	1.32	1.35	1.38
Error Rate (CV)	18.11	22.73	28.44	32.40	35.66	43.23
Error Rate (StDev)	4.25	4.14	3.88	3.06	2.26	1.71
Error Rate (Mean)	23.45	18.20	13.63	9.43	6.34	3.94
Average Separation (CV)	5.51	5.79	4.15	5.57	4.90	5.11
Average Separation (StDev)	0.62	0.66	0.48	0.64	0.56	0.60
Average Separation (Mean)	11.29	11.36	11.51	11.52	11.54	11.78
Offset	1.40	1.43	1.46	1.49	1.52	1.55
Error Rate (CV)	66.77	103.88	131.76	169.48	216.12	282.05
Error Rate (StDev)	1.67	1.30	0.78	0.58	0.29	0.18
Error Rate (Mean)	2.50	1.25	0.60	0.34	0.14	0.06
Average Separation (CV)	5.07	4.58	4.63	4.72	5.17	4.68
Average Separation (StDev)	0.59	0.54	0.55	0.57	0.63	0.57
Average Separation (Mean)	11.69	11.82	11.95	12.05	12.13	12.23
S500						
Offset	1.25	1.27	1.30	1.32	1.35	1.38
Error Rate (CV)	8.82	9.27	12.60	14.24	16.07	22.12
Error Rate (StDev)	2.04	1.73	1.68	1.34	1.00	0.89
Error Rate (Mean)	23.10	18.65	13.30	9.39	6.20	4.02
Average Separation (CV)	1.95	2.22	2.18	2.20	2.07	2.26
Average Separation (StDev)	0.22	0.25	0.25	0.25	0.24	0.26
Average Separation (Mean)	11.32	11.41	11.47	11.51	11.61	11.72
Offset	1.40	1.43	1.46	1.49	1.52	1.55
Error Rate (CV)	26.20	32.99	41.22	55.49	71.71	111.01
Error Rate (StDev)	0.61	0.41	0.29	0.20	0.15	0.11
Error Rate (Mean)	2.32	1.26	0.71	0.36	0.20	0.10
Average Separation (CV)	2.07	2.24	2.27	2.28	2.07	2.11
Average Separation (StDev)	0.24	0.26	0.27	0.27	0.25	0.26
Average Separation (Mean)	11.80	11.84	11.96	12.03	12.07	12.18
S1000						
Offset	1.25	1.27	1.30	1.32	1.35	1.38
Error Rate (CV)	5.45	6.50	8.69	10.62	12.95	18.58
Error Rate (StDev)	1.27	1.21	1.18	0.99	0.82	0.74
Error Rate (Mean)	23.39	18.55	13.60	9.31	6.34	3.96
Average Separation (CV)	1.53	1.62	1.90	1.34	1.75	1.48
Average Separation (StDev)	0.17	0.19	0.22	0.15	0.20	0.17
Average Separation (Mean)	11.34	11.40	11.47	11.51	11.60	11.67
Offset	1.40	1.43	1.46	1.49	1.52	1.55
Error Rate (CV)	19.22	25.00	29.02	38.72	62.13	68.32
Error Rate (StDev)	0.45	0.33	0.21	0.14	0.11	0.06
Error Rate (Mean)	2.35	1.32	0.72	0.36	0.18	0.09
Average Separation (CV)	1.51	1.50	1.42	1.60	1.57	1.54
Average Separation (StDev)	0.18	0.18	0.17	0.19	0.19	0.19
Average Separation (Mean)	11.75	11.86	11.90	12.05	12.11	12.18

## B.1.11 MINDEV - Case MS20 CV results

**Table B.6:** Coefficient of variation results for case MS20 using method MINDEV, of the *LDA* misclassification rates and average separation values in 100 runs of the experiment.

S100						
Offset	1.40	1.46	1.52	1.58	1.65	1.72
Error Rate (CV)	15.74	19.78	27.20	32.51	36.16	61.03
Error Rate (StDev)	5.43	5.73	5.90	4.60	3.12	3.19
Error Rate (Mean)	34.50	28.96	21.71	14.14	8.62	5.22
Average Separation (CV)	5.04	5.46	4.78	5.13	4.93	4.90
Average Separation (StDev)	0.56	0.61	0.53	0.58	0.56	0.56
Average Separation (Mean)	11.19	11.12	11.19	11.33	11.41	11.43
Offset	1.79	1.86	1.93	2.01	2.10	2.18
Error Rate (CV)	70.50	102.76	171.65	230.28	389.89	397.81
Error Rate (StDev)	1.65	1.13	0.73	0.28	0.18	0.12
Error Rate (Mean)	2.34	1.09	0.42	0.12	0.04	0.03
Average Separation (CV)	4.57	5.14	5.35	4.66	4.65	4.31
Average Separation (StDev)	0.52	0.59	0.63	0.55	0.54	0.51
Average Separation (Mean)	11.45	11.57	11.69	11.75	11.71	11.91
S500						
Offset	1.46	1.52	1.58	1.65	1.72	1.79
Error Rate (CV)	8.37	13.72	15.20	18.94	24.24	30.95
Error Rate (StDev)	2.37	3.03	2.27	1.58	1.16	0.76
Error Rate (Mean)	28.27	22.10	14.91	8.36	4.79	2.45
Average Separation (CV)	2.06	2.16	2.21	2.24	2.11	1.90
Average Separation (StDev)	0.23	0.24	0.25	0.25	0.24	0.22
Average Separation (Mean)	11.23	11.26	11.29	11.39	11.46	11.54
Offset	1.86	1.93	2.01	2.10	2.18	2.27
Error Rate (CV)	39.80	52.96	82.23	115.06	192.57	356.47
Error Rate (StDev)	0.47	0.28	0.18	0.09	0.07	0.03
Error Rate (Mean)	1.19	0.53	0.22	0.08	0.03	0.01
Average Separation (CV)	1.85	1.88	2.12	2.06	1.84	1.96
Average Separation (StDev)	0.21	0.22	0.25	0.24	0.22	0.24
Average Separation (Mean)	11.57	11.68	11.71	11.81	11.84	11.99
S1000						
Offset	1.48	1.52	1.57	1.62	1.67	1.72
Error Rate (CV)	7.24	8.26	10.23	12.49	13.37	18.98
Error Rate (StDev)	1.95	1.79	1.66	1.41	1.01	0.91
Error Rate (Mean)	26.88	21.69	16.18	11.30	7.58	4.79
Average Separation (CV)	1.68	1.58	1.46	1.48	1.40	1.60
Average Separation (StDev)	0.19	0.18	0.17	0.17	0.16	0.18
Average Separation (Mean)	11.26	11.28	11.33	11.33	11.38	11.48
Offset	1.77	1.82	1.88	1.93	1.99	2.05
Error Rate (CV)	21.67	27.19	26.58	35.36	53.15	62.05
Error Rate (StDev)	0.63	0.46	0.26	0.17	0.15	0.08
Error Rate (Mean)	2.93	1.71	0.99	0.48	0.28	0.13
Average Separation (CV)	1.33	1.39	1.36	1.49	1.39	1.39
Average Separation (StDev)	0.15	0.16	0.16	0.17	0.16	0.16
Average Separation (Mean)	11.52	11.55	11.59	11.66	11.70	11.77

## B.1.12 MINDEV - Case MS3 CV results

**Table B.7:** Coefficient of variation results for case MS3 using method MINDEV, of the *LDA mis-classification rates and average separation values* in 100 runs of the experiment.

<b>S100</b>						
Offset	2.72	2.86	3.00	3.16	3.32	3.49
Error Rate (CV)	23.34	35.89	41.15	57.62	75.15	86.72
Error Rate (StDev)	7.36	9.05	8.46	8.41	7.47	4.80
Error Rate (Mean)	31.51	25.21	20.55	14.60	9.94	5.54
Average Separation (CV)	5.17	4.43	5.17	4.78	4.42	4.50
Average Separation (StDev)	0.58	0.50	0.58	0.54	0.50	0.51
Average Separation (Mean)	11.17	11.27	11.24	11.28	11.28	11.35
Offset	3.67	3.86	4.06	4.26	4.48	4.71
Error Rate (CV)	106.07	123.14	132.44	197.92	228.58	296.36
Error Rate (StDev)	3.53	2.19	1.63	0.81	0.51	0.28
Error Rate (Mean)	3.33	1.78	1.23	0.41	0.22	0.10
Average Separation (CV)	5.03	4.96	4.65	4.79	4.29	5.07
Average Separation (StDev)	0.57	0.57	0.53	0.55	0.50	0.59
Average Separation (Mean)	11.42	11.43	11.45	11.51	11.55	11.56
<b>S500</b>						
Offset	2.86	3.00	3.16	3.32	3.49	3.67
Error Rate (CV)	13.79	18.29	24.20	34.89	42.34	56.74
Error Rate (StDev)	3.57	3.87	3.31	2.98	2.05	1.54
Error Rate (Mean)	25.88	21.16	13.70	8.55	4.83	2.72
Average Separation (CV)	2.26	2.08	2.06	2.12	2.09	1.88
Average Separation (StDev)	0.25	0.23	0.23	0.24	0.24	0.21
Average Separation (Mean)	11.23	11.22	11.34	11.32	11.39	11.38
Offset	3.86	4.06	4.26	4.48	4.71	4.95
Error Rate (CV)	65.31	81.48	76.74	106.33	111.67	208.56
Error Rate (StDev)	0.91	0.64	0.31	0.22	0.10	0.07
Error Rate (Mean)	1.39	0.79	0.40	0.20	0.09	0.04
Average Separation (CV)	2.12	1.98	1.91	1.91	2.12	2.12
Average Separation (StDev)	0.24	0.23	0.22	0.22	0.24	0.25
Average Separation (Mean)	11.44	11.51	11.50	11.57	11.55	11.59
<b>S1000</b>						
Offset	2.86	3.00	3.16	3.32	3.49	3.67
Error Rate (CV)	11.05	15.48	17.87	24.64	30.04	42.75
Error Rate (StDev)	2.89	3.08	2.51	2.08	1.47	1.13
Error Rate (Mean)	26.12	19.88	14.05	8.44	4.90	2.65
Average Separation (CV)	1.44	1.56	1.50	1.51	1.32	1.43
Average Separation (StDev)	0.16	0.18	0.17	0.17	0.15	0.16
Average Separation (Mean)	11.23	11.24	11.27	11.34	11.37	11.40
Offset	3.86	4.06	4.26	4.48	4.71	4.95
Error Rate (CV)	41.66	58.07	70.68	76.94	99.18	154.40
Error Rate (StDev)	0.59	0.39	0.24	0.12	0.07	0.06
Error Rate (Mean)	1.42	0.66	0.33	0.15	0.07	0.04
Average Separation (CV)	1.33	1.60	1.37	1.37	1.60	1.51
Average Separation (StDev)	0.15	0.18	0.16	0.16	0.19	0.18
Average Separation (Mean)	11.44	11.48	11.52	11.55	11.59	11.61

### B.1.13 MINDEV - Case MS1 CV results

**Table B.8:** Coefficient of variation results for case MS1 using method MINDEV, of the *LDA mis-classification rates and average separation values* in 100 runs of the experiment.

S100						
Offset	7.03	7.39	7.77	8.17	8.58	9.03
Error Rate (CV)	62.49	83.17	89.31	128.89	121.06	156.80
Error Rate (StDev)	12.70	11.84	10.42	7.76	4.05	3.83
Error Rate (Mean)	20.32	14.23	11.66	6.02	3.35	2.44
Average Separation (CV)	5.41	5.92	3.97	5.16	4.65	5.09
Average Separation (StDev)	0.61	0.67	0.45	0.58	0.52	0.58
Average Separation (Mean)	11.19	11.27	11.35	11.32	11.28	11.46
Offset	9.49	9.97	10.49	11.02	11.59	12.18
Error Rate (CV)	226.10	231.80	258.20	259.23	310.79	418.55
Error Rate (StDev)	4.49	2.26	0.65	1.43	0.42	0.27
Error Rate (Mean)	1.99	0.98	0.25	0.55	0.14	0.06
Average Separation (CV)	5.15	4.79	4.80	4.73	5.05	4.98
Average Separation (StDev)	0.58	0.55	0.55	0.54	0.58	0.57
Average Separation (Mean)	11.34	11.41	11.47	11.47	11.49	11.54
S500						
Offset	7.03	7.39	7.77	8.17	8.58	9.03
Error Rate (CV)	34.68	43.84	55.20	84.84	84.37	104.97
Error Rate (StDev)	7.58	6.10	5.34	4.27	2.08	1.27
Error Rate (Mean)	21.87	13.92	9.67	5.04	2.47	1.21
Average Separation (CV)	1.98	2.08	2.26	2.14	2.12	1.88
Average Separation (StDev)	0.22	0.23	0.26	0.24	0.24	0.21
Average Separation (Mean)	11.24	11.31	11.29	11.31	11.36	11.35
Offset	9.49	9.97	10.49	11.02	11.59	12.18
Error Rate (CV)	138.09	167.33	200.43	365.41	285.88	420.66
Error Rate (StDev)	0.75	0.32	0.29	0.29	0.06	0.04
Error Rate (Mean)	0.54	0.19	0.14	0.08	0.02	0.01
Average Separation (CV)	1.94	2.06	2.13	2.06	2.07	1.99
Average Separation (StDev)	0.22	0.24	0.24	0.24	0.24	0.23
Average Separation (Mean)	11.46	11.41	11.45	11.44	11.47	11.54
S1000						
Offset	7.03	7.39	7.77	8.17	8.58	9.03
Error Rate (CV)	23.04	32.47	42.59	48.07	76.36	103.27
Error Rate (StDev)	4.85	4.64	3.68	2.09	1.66	1.08
Error Rate (Mean)	21.06	14.30	8.63	4.34	2.17	1.05
Average Separation (CV)	1.82	1.45	1.41	1.55	1.56	1.62
Average Separation (StDev)	0.21	0.16	0.16	0.18	0.18	0.18
Average Separation (Mean)	11.26	11.28	11.35	11.33	11.34	11.37
Offset	9.49	9.97	10.49	11.02	11.59	12.18
Error Rate (CV)	111.34	148.35	168.18	213.10	240.40	522.23
Error Rate (StDev)	0.44	0.30	0.13	0.08	0.03	0.01
Error Rate (Mean)	0.39	0.20	0.08	0.04	0.01	0.00
Average Separation (CV)	1.21	1.40	1.34	1.58	1.34	1.36
Average Separation (StDev)	0.14	0.16	0.15	0.18	0.15	0.16
Average Separation (Mean)	11.41	11.42	11.44	11.47	11.46	11.53

## Appendix B.2

---

# MAXMEAN Results

### Contents

1. MAXMEAN - Case MS120 statistics results
2. MAXMEAN - Case MS20 statistics results
3. MAXMEAN - Case MS3 statistics results
4. MAXMEAN - Case MS1 statistics results
5. MAXMEAN - LDA Boundaries
6. MAXMEAN - Error Plots for case MS120
7. MAXMEAN - Error Plots for case MS20
8. MAXMEAN - Error Plots for case MS3
9. MAXMEAN - Error Plots for case MS1
10. MAXMEAN - Case MS120 CV results
11. MAXMEAN - Case MS20 CV results
12. MAXMEAN - Case MS3 CV results
13. MAXMEAN - Case MS1 CV results

## B.2.1 MAXMEAN - Case MS120 statistics results

**Table B.9:** Average LDA misclassification rates and average separation values for the case MS120, applying the MAXMEAN method in 100 runs of the experiment.

<b>S100</b>						
<b>Offset</b>	1.25	1.27	1.30	1.32	1.35	1.38
<b>Error Rate (%)</b>	24.39	19.35	12.87	9.70	6.54	3.59
<b>Average Separation</b>	11.28	11.42	11.41	11.56	11.67	11.67
<b>Offset</b>	1.40	1.43	1.46	1.49	1.52	1.55
<b>Error Rate (%)</b>	1.95	1.21	0.51	0.36	0.13	0.10
<b>Average Separation</b>	11.82	11.84	11.93	11.99	12.09	12.21
<b>S500</b>						
<b>Offset</b>	1.25	1.27	1.30	1.32	1.35	1.38
<b>Error Rate (%)</b>	23.59	18.17	13.67	9.37	5.96	3.74
<b>Average Separation</b>	11.37	11.39	11.49	11.51	11.61	11.63
<b>Offset</b>	1.40	1.43	1.46	1.49	1.52	1.55
<b>Error Rate (%)</b>	2.13	1.29	0.77	0.40	0.23	0.11
<b>Average Separation</b>	11.78	11.84	11.91	12.03	12.08	12.24
<b>S1000</b>						
<b>Offset</b>	1.25	1.27	1.30	1.32	1.35	1.38
<b>Error Rate (%)</b>	23.67	18.37	13.65	9.40	6.15	3.80
<b>Average Separation</b>	11.33	11.39	11.43	11.51	11.57	11.68
<b>Offset</b>	1.40	1.43	1.46	1.49	1.52	1.55
<b>Error Rate (%)</b>	2.31	1.35	0.75	0.38	0.21	0.09
<b>Average Separation</b>	11.75	11.84	11.92	12.00	12.09	12.22

Table B.9 gives the average statistics values in 100 runs of the simulation experiment for method MAXMEAN in case MS120 and offsets in the range 1.25 – 1.55. Offsets in the range 1.26 – 1.45 are required in all sample size cases, to achieve misclassification rates of  $\approx 20 - 1\%$  respectively. The average separation between the two data sets is  $\approx 11.35$  for a 20% misclassification rate and the two data sets are almost linearly separable when the average separation is  $\approx 12.2$  with *misclassification rate*  $\approx 0.1\%$ .

## B.2.2 MAXMEAN - Case MS20 statistics results

**Table B.10:** Average LDA misclassification rates and average separation values for the case MS20, applying the MAXMEAN method in 100 runs of the experiment.

<b>S100</b>						
<b>Offset</b>	1.40	1.46	1.52	1.58	1.65	1.72
<b>Error Rate (%)</b>	35.81	28.72	20.91	13.67	7.53	3.95
<b>Average Separation</b>	11.15	11.33	11.46	11.21	11.42	11.40
<b>Offset</b>	1.79	1.86	1.93	2.01	2.10	2.18
<b>Error Rate (%)</b>	2.42	1.05	0.31	0.18	0.07	0.02
<b>Average Separation</b>	11.52	11.55	11.63	11.72	11.83	11.89
<b>S500</b>						
<b>Offset</b>	1.48	1.52	1.57	1.62	1.67	1.72
<b>Error Rate (%)</b>	26.79	21.34	15.11	10.21	6.79	3.90
<b>Average Separation</b>	11.25	11.28	11.30	11.34	11.39	11.43
<b>Offset</b>	1.77	1.82	1.88	1.93	1.99	2.05
<b>Error Rate (%)</b>	2.52	1.39	0.73	0.45	0.22	0.12
<b>Average Separation</b>	11.50	11.53	11.57	11.68	11.72	11.76
<b>S1000</b>						
<b>Offset</b>	1.48	1.52	1.57	1.62	1.67	1.72
<b>Error Rate (%)</b>	27.23	21.02	15.27	10.27	6.48	3.97
<b>Average Separation</b>	11.24	11.26	11.29	11.36	11.39	11.44
<b>Offset</b>	1.77	1.82	1.88	1.93	1.99	2.05
<b>Error Rate (%)</b>	2.340	1.45	0.76	0.41	0.22	0.09
<b>Average Separation</b>	11.51	11.56	11.59	11.63	11.69	11.76

The misclassification rates and average separation values of the experiments for offsets in the range 1.40 – 2.18 for case S100 and 1.48 – 2.05 in the other two sample size cases, can be seen in Table B.10. Offsets in the range 1.52 – 1.85 are required in all sample size cases, to achieve misclassification rates of  $\approx 20 - 1\%$  respectively. The average separation between the two data sets is  $\approx 11.3$  for a 20% misclassification rate and the two data sets are almost linearly separable when the average separation is  $\approx 11.8$  with misclassification rate  $\approx 0.1\%$ .

## B.2.3 MAXMEAN - Case MS3 statistics results

**Table B.11:** Average LDA misclassification rates and average separation values for the case MS3, applying the MAXMEAN method in 100 runs of the experiment.

<b>S100</b>						
<b>Offset</b>	2.72	2.83	2.94	3.06	3.19	3.32
<b>Error Rate (%)</b>	26.79	22.11	18.66	13.96	11.17	7.87
<b>Average Separation</b>	11.21	11.15	11.30	11.33	11.38	11.32
<b>Offset</b>	3.46	3.60	3.74	3.90	4.06	4.22
<b>Error Rate (%)</b>	5.87	5.00	2.94	2.24	1.46	0.77
<b>Average Separation</b>	11.31	11.41	11.50	11.49	11.47	11.48
<b>S500</b>						
<b>Offset</b>	2.86	3.00	3.16	3.32	3.49	3.67
<b>Error Rate (%)</b>	21.25	14.86	10.88	7.50	4.59	3.18
<b>Average Separation</b>	11.24	11.27	11.35	11.34	11.37	11.41
<b>Offset</b>	3.86	4.06	4.26	4.48	4.71	4.95
<b>Error Rate (%)</b>	1.93	1.39	0.85	0.58	0.31	0.19
<b>Average Separation</b>	11.45	11.48	11.51	11.56	11.53	11.62
<b>S1000</b>						
<b>Offset</b>	2.86	3.00	3.16	3.32	3.49	3.67
<b>Error Rate (%)</b>	21.36	15.15	10.30	7.12	4.81	3.10
<b>Average Separation</b>	11.22	11.28	11.30	11.34	11.38	11.42
<b>Offset</b>	3.86	4.06	4.26	4.48	4.71	4.95
<b>Error Rate (%)</b>	2.05	1.34	0.82	0.46	0.32	0.18
<b>Average Separation</b>	11.43	11.46	11.49	11.56	11.57	11.61

Table B.11 gives the misclassification rates and average separation values of the experiments, for offsets in the range 2.72 – 4.22 for case S100 and 2.86 – 4.95 for the other two sample size cases. Offsets in the range 2.9 – 4.15 are required, to achieve misclassification rates of  $\approx 20 - 1\%$  respectively. The average separation between the two data sets is  $\approx 11.25$  for a 20% misclassification rate and the two data sets are almost linearly separable when the average separation is  $\approx 11.6$  with misclassification rate  $\approx 0.2\%$ .



## B.2.4 MAXMEAN - Case MS1 statistics results

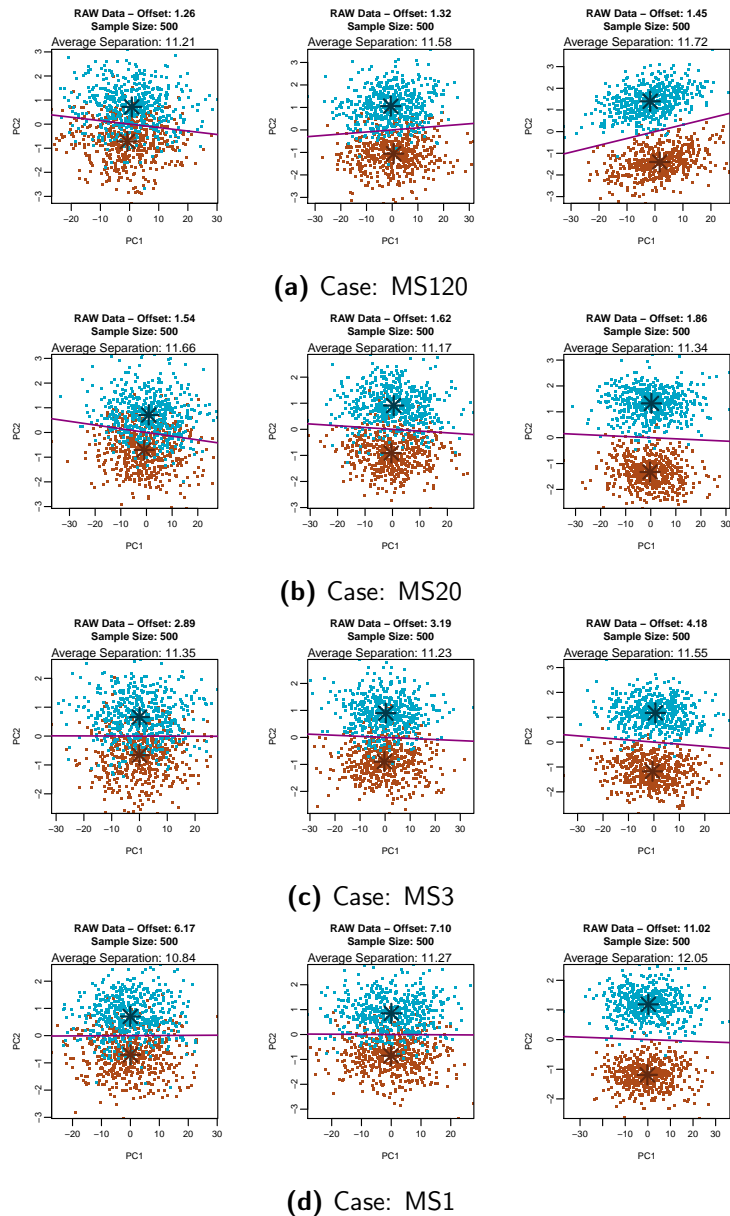
**Table B.12:** Average LDA misclassification rates and average separation values for the case MS1, applying the MAXMEAN method in 100 runs of the experiment.

<b>S100</b>						
<b>Offset</b>	5.47	6.05	6.69	7.39	8.17	9.03
<b>Error Rate (%)</b>	30.95	24.48	15.23	11.95	6.20	3.65
<b>Average Separation</b>	11.13	11.23	11.23	11.33	11.42	11.39
<b>Offset</b>	9.97	11.02	12.18	13.46	14.88	16.44
<b>Error Rate (%)</b>	2.16	1.13	0.61	0.30	0.19	0.09
<b>Average Separation</b>	11.48	11.48	11.44	11.51	11.62	11.65
<b>S500</b>						
<b>Offset</b>	5.47	6.05	6.69	7.39	8.17	9.03
<b>Error Rate (%)</b>	32.21	22.92	14.18	8.45	5.00	2.82
<b>Average Separation</b>	11.21	11.26	11.24	11.27	11.37	11.38
<b>Offset</b>	9.97	11.02	12.18	13.46	14.88	16.44
<b>Error Rate (%)</b>	1.99	1.10	0.57	0.37	0.18	0.12
<b>Average Separation</b>	11.44	11.41	11.49	11.54	11.56	11.61
<b>S1000</b>						
<b>Offset</b>	5.47	6.05	6.69	7.39	8.17	9.03
<b>Error Rate (%)</b>	31.99	22.32	13.54	8.19	4.98	2.88
<b>Average Separation</b>	11.18	11.24	11.27	11.34	11.33	11.37
<b>Offset</b>	9.97	11.02	12.18	13.46	14.88	16.44
<b>Error Rate (%)</b>	1.80	1.03	0.55	0.31	0.15	0.08
<b>Average Separation</b>	11.43	11.47	11.48	11.56	11.58	11.62

Table B.12 gives the misclassification rates and average separation values of the experiments, for offsets in the range 5.47 – 16.44 for all the sample size cases. Offsets in the range 6.3 – 11.2 are required, to achieve misclassification rates of  $\approx 20 - 1\%$  respectively. The average separation between the two data sets is  $\approx 11.2$  for a 20% misclassification rate and the two data sets are almost linearly separable when the average separation is  $\approx 11.6$  with misclassification rate  $\approx 0.1\%$ .

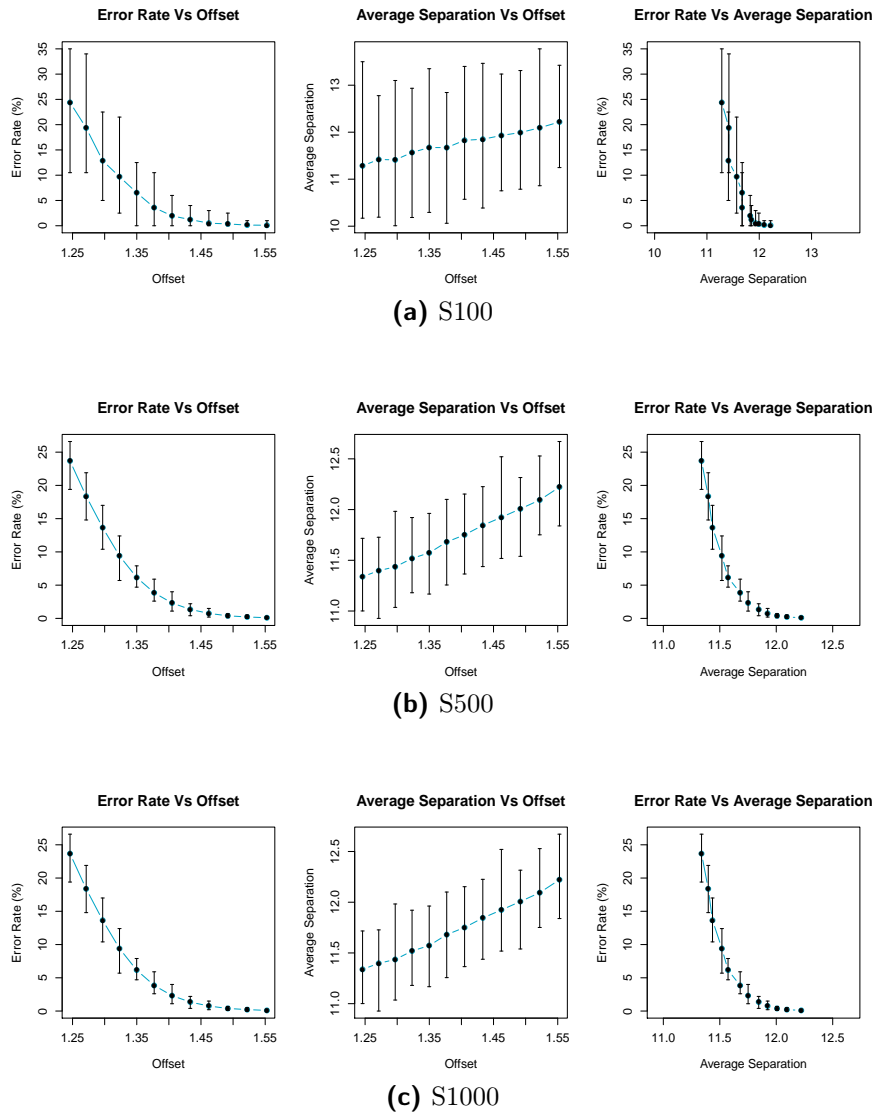
## B.2.5 MAXMEAN - LDA Boundaries

An illustration of how mean-shifting affects the capability of PCA to discriminate the two data sets in all four MS cases with S500 for MAXMEAN, superimposed with the LDA boundary for the two artificial data sets, can be seen in Figure B.6, for suitably selected offsets which correspond to 20 %, 10 % and 1 % misclassification rates respectively.



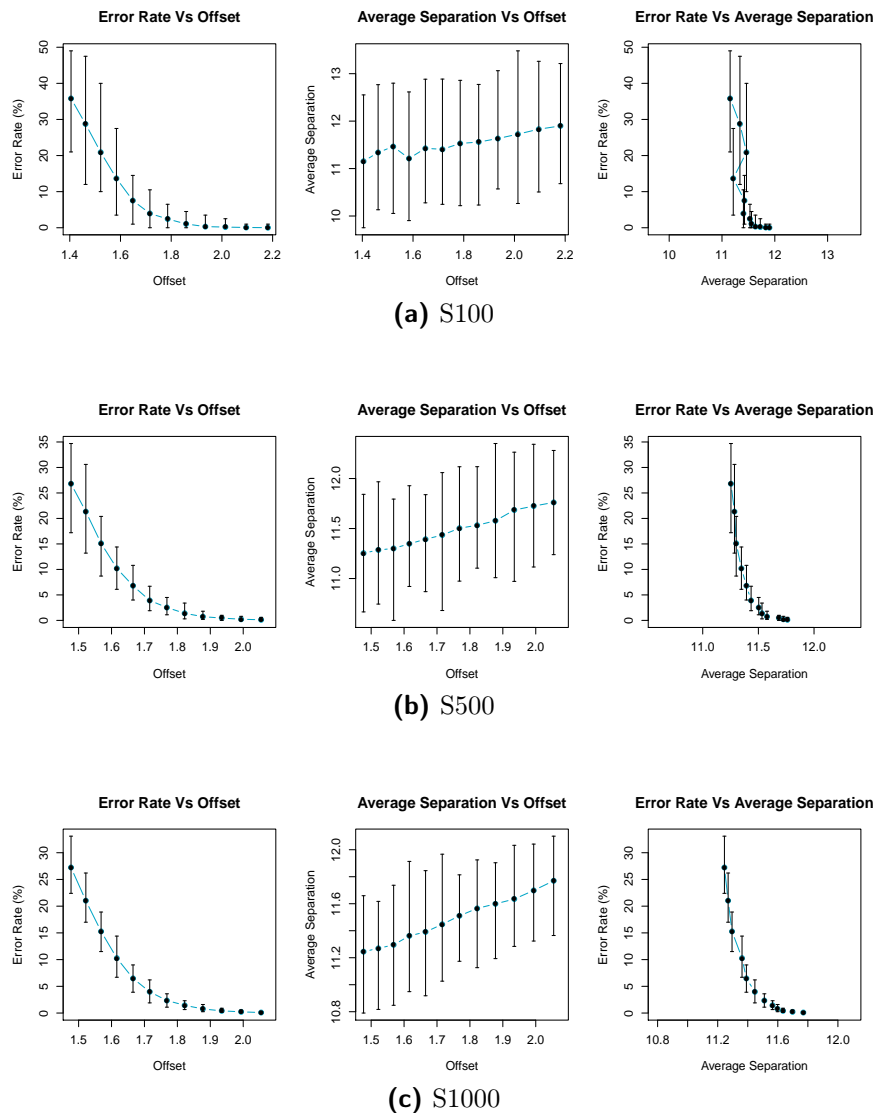
**Figure B.6:** Visualisation of the LDA boundaries for the two artificial data sets in all four MS cases (MAXMEAN). The data corresponds to the first two PCs for LDA. The reference and test data points are depicted in brown and blue respectively.

## B.2.6 MAXMEAN - Error Plots for case MS120



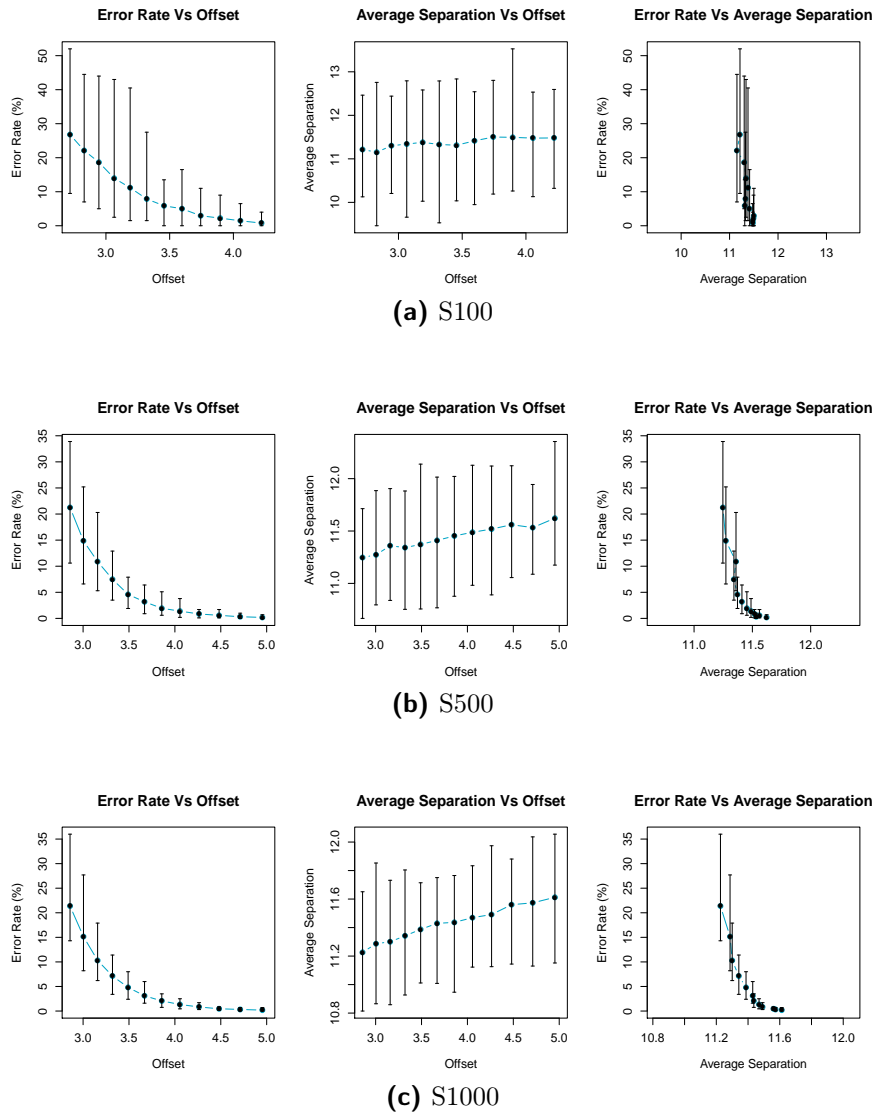
**Figure B.7:** Visualisation of the relation among *LDA misclassification rates*, *average separation* and *offsets* in the case MS120 with MAXMEAN. The blue lines represent the mean values of each statistic for each offset in the selected offset range. The offsets are the multiplicative factors on the original scale of the data. The vertical error bars are such that the top and bottom of a bar correspond to the maximum and minimum statistic value at the respective offset. The statistics values are the average values in 100 runs of the experiment. The two average separation plots in each sample size case are drawn using the same range of values for the *average separation*.

## B.2.7 MAXMEAN - Error Plots for case MS20



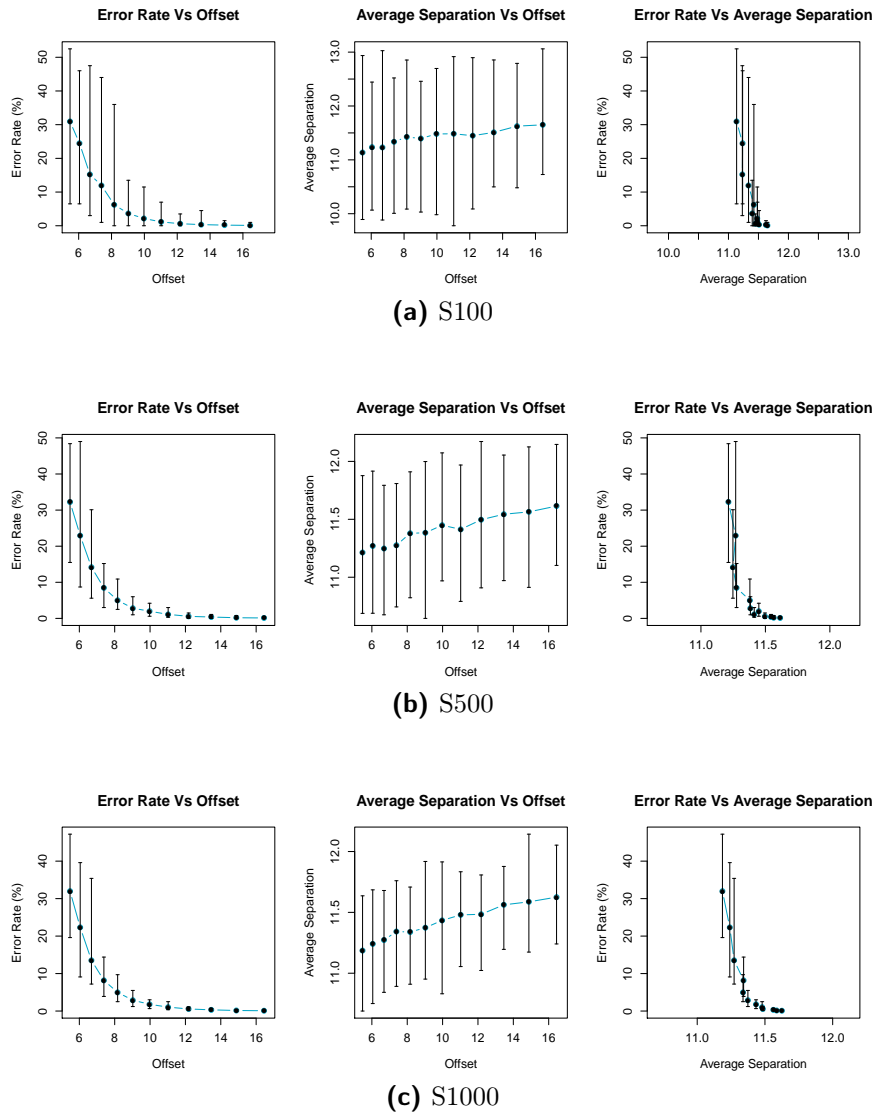
**Figure B.8:** Visualisation of the relation among *LDA misclassification rates*, *average separation* and *offsets* in the case MS20 with MAXMEAN. The blue lines represent the mean values of each statistic for each offset in the selected offset range. The offsets are the multiplicative factors on the original scale of the data. The vertical error bars are such that the top and bottom of a bar correspond to the maximum and minimum statistic value at the respective offset. The statistics values are the average values in 100 runs of the experiment. The two average separation plots in each sample size case are drawn using the same range of values for the *average separation*.

## B.2.8 MAXMEAN - Error Plots for case MS3



**Figure B.9:** Visualisation of the relation among *LDA misclassification rates*, *average separation* and *offsets* in the case MS3 with MAXMEAN. The blue lines represent the mean values of each statistic for each offset in the selected offset range. The offsets are the multiplicative factors on the original scale of the data. The vertical error bars are such that the top and bottom of a bar correspond to the maximum and minimum statistic value at the respective offset. The statistics values are the average values in 100 runs of the experiment. The two average separation plots in each sample size case are drawn using the same range of values for the *average separation*.

## B.2.9 MAXMEAN - Error Plots for case MS1



**Figure B.10:** Visualisation of the relation among *LDA misclassification rates*, *average separation* and *offsets* in the case MS1 with MAXMEAN. The blue lines represent the mean values of each statistic for each offset in the selected offset range. The offsets are the multiplicative factors on the original scale of the data. The vertical error bars are such that the top and bottom of a bar correspond to the maximum and minimum statistic value at the respective offset. The statistics values are the average values in 100 runs of the experiment. The two average separation plots in each sample size case are drawn using the same range of values for the *average separation*.

## B.2.10 MAXMEAN - Case MS120 CV results

**Table B.13:** Coefficient of variation results for case MS120 using method MAXMEAN, of the *LDA* misclassification rates and average separation values in 100 runs of the experiment.

<b>S100</b>						
Offset	1.25	1.27	1.30	1.32	1.35	1.38
Error Rate (CV)	19.30	23.45	26.86	34.36	37.76	54.99
Error Rate (StDev)	4.71	4.54	3.46	3.33	2.47	1.98
Error Rate (Mean)	24.39	19.35	12.87	9.70	6.54	3.60
Average Separation (CV)	4.80	4.68	4.55	5.36	5.11	4.72
Average Separation (StDev)	0.54	0.53	0.52	0.62	0.60	0.55
Average Separation (Mean)	11.28	11.42	11.41	11.57	11.67	11.67
Offset	1.40	1.43	1.46	1.49	1.52	1.55
Error Rate (CV)	75.30	95.26	129.05	153.01	229.47	213.20
Error Rate (StDev)	0.58	0.60	0.48	0.58	0.52	0.57
Error Rate (Mean)	1.96	1.22	0.52	0.36	0.13	0.10
Average Separation (CV)	4.88	5.05	4.06	4.88	4.32	4.65
Average Separation (StDev)	0.58	0.60	0.48	0.58	0.52	0.57
Average Separation (Mean)	11.83	11.85	11.93	11.99	12.09	12.22
<b>S500</b>						
Offset	1.25	1.27	1.30	1.32	1.35	1.38
Error Rate (CV)	8.86	10.88	11.89	15.35	17.61	24.64
Error Rate (StDev)	2.09	1.98	1.63	1.44	1.05	0.92
Error Rate (Mean)	23.59	18.18	13.68	9.38	5.96	3.75
Average Separation (CV)	2.21	1.85	2.19	2.34	2.23	2.09
Average Separation (StDev)	0.25	0.21	0.25	0.27	0.26	0.24
Average Separation (Mean)	11.37	11.40	11.49	11.51	11.62	11.64
Offset	1.40	1.43	1.46	1.49	1.52	1.55
Error Rate (CV)	31.37	34.37	42.59	58.05	74.52	90.88
Error Rate (StDev)	0.67	0.45	0.33	0.24	0.17	0.10
Error Rate (Mean)	2.14	1.30	0.78	0.41	0.23	0.11
Average Separation (CV)	2.29	2.09	1.95	2.06	2.17	2.07
Average Separation (StDev)	0.27	0.25	0.23	0.25	0.26	0.25
Average Separation (Mean)	11.78	11.85	11.91	12.03	12.09	12.25
<b>S1000</b>						
Offset	1.25	1.27	1.30	1.32	1.35	1.38
Error Rate (CV)	5.82	8.50	8.72	11.71	12.85	17.67
Error Rate (StDev)	1.38	1.56	1.19	1.10	0.79	0.68
Error Rate (Mean)	23.67	18.37	13.65	9.40	6.16	3.85
Average Separation (CV)	1.43	1.40	1.77	1.39	1.66	1.37
Average Separation (StDev)	0.16	0.16	0.20	0.16	0.19	0.16
Average Separation (Mean)	11.34	11.40	11.44	11.52	11.57	11.68
Offset	1.40	1.43	1.46	1.49	1.52	1.55
Error Rate (CV)	19.69	25.28	32.46	35.75	51.38	67.48
Error Rate (StDev)	0.46	0.34	0.25	0.14	0.11	0.07
Error Rate (Mean)	2.32	1.35	0.76	0.39	0.22	0.10
Average Separation (CV)	1.48	1.33	1.57	1.40	1.34	1.45
Average Separation (StDev)	0.17	0.16	0.19	0.17	0.16	0.18
Average Separation (Mean)	11.75	11.84	11.92	12.01	12.09	12.22

## B.2.11 MAXMEAN - Case MS20 CV results

**Table B.14:** Coefficient of variation results for case MS20 using method MAXMEAN, of the *LDA* misclassification rates and average separation values in 100 runs of the experiment.

S100						
Offset	1.40	1.46	1.52	1.58	1.65	1.72
Error Rate (CV)	17.15	25.50	31.82	38.90	42.53	58.26
Error Rate (StDev)	6.14	7.33	6.65	5.32	3.20	2.30
Error Rate (Mean)	35.81	28.73	20.91	13.67	7.53	3.96
Average Separation (CV)	5.00	4.54	4.87	5.32	4.92	4.81
Average Separation (StDev)	0.56	0.51	0.56	0.60	0.56	0.55
Average Separation (Mean)	11.15	11.34	11.46	11.21	11.43	11.41
Offset	1.79	1.86	1.93	2.01	2.10	2.18
Error Rate (CV)	70.11	109.24	200.09	247.93	305.21	607.20
Error Rate (StDev)	1.70	1.15	0.63	0.46	0.23	0.12
Error Rate (Mean)	2.42	1.05	0.32	0.18	0.08	0.02
Average Separation (CV)	4.71	4.37	4.39	4.69	4.06	4.85
Average Separation (StDev)	0.54	0.51	0.51	0.55	0.48	0.58
Average Separation (Mean)	11.53	11.56	11.63	11.73	11.83	11.90
S500						
Offset	1.48	1.52	1.57	1.62	1.67	1.72
Error Rate (CV)	12.75	13.72	17.82	16.53	21.19	24.96
Error Rate (StDev)	3.42	2.93	2.69	1.69	1.44	0.98
Error Rate (Mean)	26.80	21.34	15.12	10.21	6.79	3.91
Average Separation (CV)	1.87	2.26	2.27	1.97	2.16	2.09
Average Separation (StDev)	0.21	0.25	0.26	0.22	0.25	0.24
Average Separation (Mean)	11.25	11.29	11.30	11.35	11.39	11.43
Offset	1.77	1.82	1.88	1.93	1.99	2.05
Error Rate (CV)	27.56	42.41	44.08	52.16	65.77	105.26
Error Rate (StDev)	0.70	0.59	0.32	0.24	0.15	0.13
Error Rate (Mean)	2.53	1.40	0.73	0.46	0.23	0.12
Average Separation (CV)	2.08	1.77	2.37	2.11	2.11	1.92
Average Separation (StDev)	0.24	0.20	0.27	0.25	0.25	0.23
Average Separation (Mean)	11.50	11.53	11.58	11.68	11.72	11.76
S1000						
Offset	1.48	1.52	1.57	1.62	1.67	1.72
Error Rate (CV)	8.82	9.35	10.47	15.62	15.66	21.00
Error Rate (StDev)	2.40	1.97	1.60	1.60	1.02	0.83
Error Rate (Mean)	27.24	21.02	15.28	10.28	6.48	3.97
Average Separation (CV)	1.56	1.47	1.59	1.52	1.44	1.61
Average Separation (StDev)	0.18	0.17	0.18	0.17	0.16	0.18
Average Separation (Mean)	11.24	11.27	11.30	11.36	11.39	11.45
Offset	1.77	1.82	1.88	1.93	1.99	2.05
Error Rate (CV)	21.48	24.24	35.17	36.75	48.78	70.10
Error Rate (StDev)	0.50	0.35	0.27	0.15	0.11	0.07
Error Rate (Mean)	2.34	1.45	0.76	0.42	0.22	0.09
Average Separation (CV)	1.33	1.43	1.49	1.39	1.41	1.39
Average Separation (StDev)	0.15	0.17	0.17	0.16	0.17	0.16
Average Separation (Mean)	11.51	11.57	11.60	11.63	11.70	11.77



## B.2.12 MAXMEAN - Case MS3 CV results

**Table B.15:** Coefficient of variation results for case MS3 using method MAXMEAN, of the *LDA* misclassification rates and average separation values in 100 runs of the experiment.

<b>S100</b>						
Offset	2.72	2.83	2.94	3.06	3.19	3.32
Error Rate (CV)	35.47	38.29	46.84	54.05	63.68	66.72
Error Rate (StDev)	9.50	8.47	8.74	7.55	7.12	5.25
Error Rate (Mean)	26.80	22.11	18.66	13.96	11.18	7.87
Average Separation (CV)	4.34	4.96	4.46	4.59	4.13	5.15
Average Separation (StDev)	0.49	0.55	0.50	0.52	0.47	0.58
Average Separation (Mean)	11.21	11.15	11.30	11.34	11.38	11.33
Offset	3.46	3.60	3.74	3.90	4.06	4.22
Error Rate (CV)	57.56	71.87	84.49	93.34	95.47	113.93
Error Rate (StDev)	3.38	3.59	2.49	2.09	1.39	0.88
Error Rate (Mean)	5.87	5.00	2.94	2.24	1.46	0.78
Average Separation (CV)	4.95	4.32	4.42	4.77	4.74	4.37
Average Separation (StDev)	0.56	0.49	0.51	0.55	0.54	0.50
Average Separation (Mean)	11.31	11.41	11.50	11.49	11.48	11.48
<b>S500</b>						
Offset	2.86	3.00	3.16	3.32	3.49	3.67
Error Rate (CV)	22.56	24.47	25.49	25.11	29.31	35.34
Error Rate (StDev)	4.80	3.64	2.77	1.88	1.35	1.12
Error Rate (Mean)	21.26	14.86	10.88	7.51	4.60	3.18
Average Separation (CV)	2.07	2.00	1.91	1.95	2.22	2.19
Average Separation (StDev)	0.23	0.23	0.22	0.22	0.25	0.25
Average Separation (Mean)	11.25	11.27	11.36	11.34	11.37	11.41
Offset	3.86	4.06	4.26	4.48	4.71	4.95
Error Rate (CV)	35.90	44.38	47.14	52.14	69.41	80.40
Error Rate (StDev)	0.70	0.62	0.40	0.31	0.22	0.15
Error Rate (Mean)	1.94	1.40	0.85	0.59	0.32	0.19
Average Separation (CV)	2.15	2.03	2.11	2.05	1.93	1.96
Average Separation (StDev)	0.25	0.23	0.24	0.24	0.22	0.23
Average Separation (Mean)	11.45	11.49	11.52	11.56	11.53	11.62
<b>S1000</b>						
Offset	2.86	3.00	3.16	3.32	3.49	3.67
Error Rate (CV)	15.03	23.20	21.58	21.20	20.27	22.07
Error Rate (StDev)	3.21	3.52	2.22	1.51	0.98	0.68
Error Rate (Mean)	21.36	15.15	10.30	7.12	4.81	3.10
Average Separation (CV)	1.69	1.68	1.44	1.47	1.28	1.32
Average Separation (StDev)	0.19	0.19	0.16	0.17	0.15	0.15
Average Separation (Mean)	11.23	11.29	11.30	11.34	11.39	11.43
Offset	3.86	4.06	4.26	4.48	4.71	4.95
Error Rate (CV)	27.88	28.32	32.69	35.55	42.37	63.01
Error Rate (StDev)	0.57	0.38	0.27	0.16	0.14	0.12
Error Rate (Mean)	2.06	1.34	0.83	0.46	0.32	0.19
Average Separation (CV)	1.54	1.40	1.51	1.48	1.56	1.42
Average Separation (StDev)	0.18	0.16	0.17	0.17	0.18	0.16
Average Separation (Mean)	11.44	11.47	11.49	11.56	11.57	11.61

## B.2.13 MAXMEAN - Case MS1 CV results

**Table B.16:** Coefficient of variation results for case MS1 using method MAXMEAN, of the *LDA* misclassification rates and average separation values in 100 runs of the experiment.

S100						
Offset	5.47	6.05	6.69	7.39	8.17	9.03
Error Rate (CV)	34.73	43.49	54.35	71.61	76.01	79.60
Error Rate (StDev)	10.75	10.65	8.28	8.56	4.72	2.91
Error Rate (Mean)	30.95	24.48	15.23	11.96	6.21	3.65
Average Separation (CV)	4.70	4.44	4.42	5.27	5.04	4.46
Average Separation (StDev)	0.52	0.50	0.50	0.60	0.58	0.51
Average Separation (Mean)	11.14	11.24	11.23	11.33	11.42	11.40
Offset	9.97	11.02	12.18	13.46	14.88	16.44
Error Rate (CV)	94.72	129.36	137.15	233.16	214.15	266.54
Error Rate (StDev)	2.05	1.47	0.84	0.77	0.41	0.25
Error Rate (Mean)	2.16	1.14	0.61	0.33	0.19	0.10
Average Separation (CV)	4.88	5.14	4.08	4.83	4.25	4.54
Average Separation (StDev)	0.56	0.59	0.47	0.56	0.49	0.53
Average Separation (Mean)	11.48	11.48	11.45	11.51	11.63	11.65
S500						
Offset	5.47	6.05	6.69	7.39	8.17	9.03
Error Rate (CV)	22.19	33.34	32.68	32.57	30.95	32.42
Error Rate (StDev)	7.15	7.65	4.64	2.75	1.55	0.91
Error Rate (Mean)	32.21	22.93	14.18	8.45	5.00	2.82
Average Separation (CV)	2.05	2.32	2.19	1.94	2.04	2.15
Average Separation (StDev)	0.23	0.26	0.25	0.22	0.23	0.24
Average Separation (Mean)	11.21	11.27	11.25	11.28	11.38	11.38
Offset	9.97	11.02	12.18	13.46	14.88	16.44
Error Rate (CV)	34.65	46.55	49.66	65.95	92.02	94.35
Error Rate (StDev)	0.69	0.52	0.28	0.25	0.17	0.12
Error Rate (Mean)	2.00	1.11	0.57	0.38	0.19	0.12
Average Separation (CV)	2.11	1.84	2.27	1.99	2.23	2.07
Average Separation (StDev)	0.24	0.21	0.26	0.23	0.26	0.24
Average Separation (Mean)	11.45	11.41	11.50	11.54	11.56	11.61
S1000						
Offset	5.47	6.05	6.69	7.39	8.17	9.03
Error Rate (CV)	19.12	25.65	26.47	26.41	24.37	25.11
Error Rate (StDev)	6.12	5.73	3.58	2.16	1.21	0.72
Error Rate (Mean)	32.00	22.33	13.54	8.19	4.98	2.89
Average Separation (CV)	1.54	1.47	1.67	1.60	1.39	1.51
Average Separation (StDev)	0.17	0.17	0.19	0.18	0.16	0.17
Average Separation (Mean)	11.19	11.24	11.27	11.34	11.34	11.37
Offset	9.97	11.02	12.18	13.46	14.88	16.44
Error Rate (CV)	27.92	31.80	39.47	43.60	62.14	83.48
Error Rate (StDev)	0.50	0.33	0.22	0.14	0.09	0.07
Error Rate (Mean)	1.81	1.04	0.56	0.32	0.15	0.08
Average Separation (CV)	1.48	1.32	1.34	1.38	1.54	1.39
Average Separation (StDev)	0.17	0.15	0.15	0.16	0.18	0.16
Average Separation (Mean)	11.43	11.48	11.48	11.56	11.59	11.63

# Appendix C

---

## R-Code

### Contents

1. **Appendix C.1** - List of R functions used in the project
2. **Appendix C.2** - Simulation Algorithm

# Appendix C.1

---

## List of R functions

### Contents

1. List of R functions

**Table C.1:** List of R functions written for the analyses in the thesis.

CHAPTER	OBJECT	FUNCTION
<b>PART I - Project and Data Description</b>		
Project Description	Figure 2.3 [2.4.3] <sup>a</sup> Internal function <sup>b</sup>	plotClinChars() <sup>b</sup> plotBars()
Pre-processing and pre-treatment of the data	Figure 4.1 [4.3.1] Figure 4.2 [4.3.2] Figures 4.3 - 4.6 [4.4.2-3] Internal function <sup>c</sup> Internal function <sup>d</sup>	compareBinWidths() <sup>1</sup> plotBaseline() <sup>c,d</sup> compareMethods() <sup>c,d</sup> createPlotEnv() plotLines()
<b>PART II - Pattern Recognition</b>		
Principal Components Analysis	Calculates Normalised Entropy [5.3.2] Calculates the Gleason-Staelin statistic [5.3.2] Figure 5.2, Table 5.2 [5.3.3] Figure 5.11 [5.3.4] Figures 5.3, 5.6-10 [5.3.4] Main function Figure 5.4 [5.3.4]	calculateNE() calculateGS() plotStopRules() plotLoadsVsPCs() <sup>c,d</sup> plotPCA() runPCA() calculateDist()
Multidimensional Scaling	Figures 6.1-7, Table 6.1 [6.4.2] Internal function <sup>e</sup> Internal function <sup>f</sup> Figure 6.8-14 [6.4.3] Internal function <sup>g</sup> Internal function <sup>h</sup>	runClassicalMDS() <sup>e,f</sup> plotMDS() plotMST() runSammonMDS() <sup>g,h</sup> plotNLM() assessQualNLM()
Cluster Analysis		
General Modules	Generic function <sup>a</sup> Method - Figure 7.29 [7.7.2.6] Main function - Figures 7.16-18,22,34 Main function <sup>j</sup> - Figures 7.4-7,14-15,20-21 Main function - Tables 7.6-8,10,12,18-23 Main function - Tables 7.9,11,13-14,24	createPLT() plotGroupMeans() <sup>a,1</sup> plotScores() plotSilhouette() tabulateChar() runChiSquareTest() <sup>a</sup>
Hierarchical Clustering	Main function Method - Figures 7.9-10 Main function - Figures 7.8,11-13 Main function Internal function <sup>i</sup> - Table 7.3 Internal function <sup>k</sup> - Figure 7.2 Internal function <sup>l</sup> - Tables 7.4-5, Figure 7.3	runHCA() plotHCAtree() <sup>a</sup> plotHCAscores() assessQualHCA() <sup>i,j,k,l</sup> calculateAC() plotBanner() computeCophCor()
Fuzzy Clustering	Main function - Tables 7.15-17	runFuzzyClust()
Hard Clustering	Main function Main function - Figure 7.19	runKmeans() computeClustInd()
Competitive Learning	Main function Method - Figures [7.5.2.5-6]	createSOM() plotSOM() <sup>a,a</sup>
<b>PART III - Data Simulation</b>		
Data Simulation	Figure 8.1 [8.2.3] Figure 8.3 [8.3.1] Figure 8.4 [8.3.2] Figure 8.5 [8.3.4] Figure 8.6 [8.3.4] Figure 8.7 [8.3.5] Internal function <sup>1</sup> Internal function <sup>2</sup> Internal function <sup>3</sup> Simulation Algorithm - Main Simulation Algorithm - Main Simulation Algorithm - Main Simulation Algorithm - Main Simulation Algorithm - Main Simulation Algorithm - Main Simulation Algorithm - Main Simulation Algorithm - Main Simulation Algorithm - Internal <sup>4</sup> Simulation Algorithm - Internal <sup>5</sup> Simulation Algorithm - Internal <sup>6</sup> Simulation Algorithm - Internal <sup>7</sup>	plotLDA() plotElemTransComp() <sup>1,3</sup> plotCovComp() <sup>1</sup> plotMS244Scaling() <sup>1</sup> plotRowScaling() <sup>2</sup> plotMeanCentring() <sup>2</sup> plotMeanSpectrum() plotSpectra() plotScores() createDataClass() generateSet() simulateData() <sup>4</sup> runSimulation() plotMeanShifting() <sup>1</sup> plotBoundaries() plotSimStats() <sup>7</sup> computeCV() plotData() <sup>1,3,5,6</sup> plotVarExpl() plotLoads() createStatsPlots()

## Appendix C.2

---

### Chapter 8 - Simulation Algorithm

#### Contents

1. `createDataClass()`
2. `generateSet()`
3. `simulateData()`
4. `runSimulation()`
5. `plotMeanShifting()`
6. `plotBoundaries()`
7. `plotSimStats()`
8. `computeCV()`
9. `plotData()`
10. `plotVarExpl()`
11. `plotLoads()`
12. `createStatsPlots()`

## C.2 R code used in the simulation algorithm

### C.2.1 Main function: createDataClass()

```
#-----#
# AIM #
# Creates the data set to be used in the simulation experiments, #
# from the original epilepsy data. #
#-----#
# ARGUMENTS #
# sFile: A character string giving the filename of the file #
# containing the spectral information. By default, the #
# file should be in the current working directory. #
# cFile: A character string giving the filename of the file #
# containing the clinical information. By default, the #
# file should be in the current working directory #
# dppm: The number of downfield variables to exclude from #
# the data e.g. dppm = 26 means, that the first 25 #
# variables (corresponding to the variables with #
# chemical shifts above 9.98 p.p.m.) will be excluded #
# from the analyses. #
# uppm: The number of upfield variables to exclude from the #
# data e.g. uppm = 275 means, that all variables with #
# chemical shifts below 0.02 p.p.m. will be excluded #
# from the analyses. #
# water: A vector giving the range of water variables to #
# exclude from the data e.g. 153:158 for the spectral #
# data with 0.04 bin width. #
# fVar: The position of the last variable in the spectral #
# data e.g. 338, in the spectral data with 0.04 bin. #
# rowScal: A logical value. If TRUE, row-scaling is applied. #
# posDev: A logical value. If TRUE, the covariance matrix is #
# converted to positive definite. #
# tol: Tolerance level for singular values and for absolute #
# eigenvalues. #
# trnType: Character string corresponding to the selection of #
# "log" or "sqrt" element transformation. #
#-----#
# DETAILS #
# -createDataClass() is also used to combine the spectral and #
# clinical information of the patients to a data frame in R, #
# in most of the statistical analyses in the thesis. #
# -createDataClass() depends on package *corpcor*, as it uses #
# the R function make.positive.definite() to convert the #
# covariance matrix of the data, to positive definite. #
#-----#

createDataClass <- function(sFile, cFile,
                           dppm, uppm, water, fVar,
                           rowScal = c(TRUE, FALSE),
                           posDev = c(TRUE, FALSE), tol,
                           trnType = c("log", "sqrt"))
{
  # Locate the epilepsy data in the storage media
```

```

specFile <- paste(getwd(), "/", sFile, sep = "")

# Exclude unsuitable for the analyses patients
exclRows <- c(23, 85, 86)

if (identical(dppm, 0))
  # Exclude the water, the high-frequency 1H NMR chemical
  # shifts above 10 p.p.m. and the low-frequency 1H NMR
  # chemical shifts below 0 p.p.m.
  exclCols <- c(1:25, water, (uppm + 1):fVar)
else if (identical(dppm, -1))
  # Retain the variables in spectral range 6-0 p.p.m.
  exclCols <- c(1:125, water, (uppm + 1):fVar)
else
  # Exclude the high-frequency 1H NMR chemical
  # shifts above (dppm - 1) p.p.m. and the low-frequency
  # 1H NMR chemical shifts below 0 p.p.m.
  exclCols <- c(1:(dppm - 1), (uppm + 1):fVar)

# Input the spectral information to R
specData <- read.csv(specFile, header = TRUE, row.names = 1,
  sep = ",")[-exclRows, -exclCols]
rownames(specData) <- sub("MN05-", "", rownames(specData))

for (i in seq(along = specData)) {
  # Remove the 'd..' and 'd.' from the rownames
  nameTwoDots <- (regexpr("d..", names(specData)[i],
    fixed = TRUE)[1] == 1)
  if (nameTwoDots)
    names(specData)[i] <- gsub("d..", "_",
      names(specData)[i])
  else names(specData)[i] <- gsub("d.", "",
    names(specData)[i])
}

if (identical(rowScal, TRUE)) {
  # Row-scale to a constant total the data
  varsTotal <- apply(specData, 1, sum)
  specData <- specData / varsTotal
  print("Row scaling to a constant total!")
}
else print("No row scaling to a constant total!")

if (missing(trnType))
  # No element transformation of the data
  print(paste("No element transformation of the data!",
    sep = ""))
else {
  if (identical(trnType, "log"))
    # Log transformation of the data
    specData <- log(specData)
  else
    # Square root transformation of the data
    specData <- sqrt(specData)

  print(paste("Element ", trnType,

```



```

        " transformation of the data!", sep = ""))
}

# Exclude unwanted clinical characteristics
exclCols <- c(8:10, 13:14, 17:18)

# Input the clinical characteristics information to R
clinFile <- paste(getwd(), "/", cFile, sep = "")
clinData <- read.csv(clinFile, header = TRUE, row.names = 1,
                    sep = ",")[-exclRows, -exclCols]

# Join the spectral with the clinical information
epilData <- list(clinical = clinData, spectra = specData)

# Extract the responders and non-responders to AEDs from the data
outcome <- epilData$clinical$Out.6.m.s
respLevels <- levels(as.factor(outcome))
selectResp <- function(x) subset(x, (outcome != respLevels[3]))
respData <- lapply(epilData, selectResp)

# Obtain the mean, standard deviation and median of the
# distributions of the variables in the extracted data
respDataMean <- apply(respData$spectra, 2, mean)
respDataStDev <- apply(respData$spectra, 2, sd)
respDataMedian <- apply(respData$spectra, 2, median)

# Calculate the covariance matrix of the extracted data and
# convert it to positive definite
if (identical(dppm, 0))
  respDataCov <- cov(respData$spectra)
else respDataCov <- cov(respData$spectra[, -c(water)])
if (posDev) respDataCov <- make.positive.definite(respDataCov,
                                                  tol)

# Construct the required data object
refData <- list(response = respData,
               respMean = respDataMean,
               respCov = respDataCov,
               respStDev = respDataStDev,
               respMedian = respDataMedian)
class(refData) <- "epiData"

return(refData)
}

```

## C.2.2 Main function: generateSet()

```
#-----#
# AIM #
# Generates a reference or test data set for the simulation #
# experiments. #
#-----#
# ARGUMENTS #
# dSet: A data set of class epiData created by #
# createDataClass(). #
# water: A vector giving the range of water variables to #
# exclude from the data e.g. 153:158 for the spectral #
# data with 0.04 bin width. #
# dppm: The number of downfield variables to exclude from #
# the data e.g. dppm = 26 means, that the first 25 #
# variables (corresponding to the variables with #
# chemical shifts above 9.98 p.p.m.) will be excluded #
# from the analyses. #
# offSet: A log value to add to the means of the selected #
# variables for mean-shifting, in the test set. To #
# generate a reference data set, an offset of size 0 #
# is used. #
# nVars: The number of variables to mean-shift. The number is #
# set to 0 when a reference set is generated. #
# nRows: The number of samples in the reference or the test #
# set. #
# vMethod: Character string corresponding to the method to #
# select the variables for mean-shifting. Three #
# methods are available: stdev, mean and median. #
# vOrder: A logical value. If TRUE, the variables are #
# selected in decreasing order, otherwise in #
# increasing order. #
# rSet: Character string corresponding to a generated #
# reference set and is used for the generation of the #
# test set. #
#-----#
# DETAILS #
# -generateSet() depends on package *MASS* to generate the #
# multivariate normal distribution samples (using mvrnorm()). #
#-----#

generateSet <- function(dSet, water, dppm, offSet, nVars, nRows,
                        vMethod = c("stdev", "mean", "median"),
                        vOrder = c(TRUE, FALSE), rSet)
{
  # if true, water variables have been removed
  noWater <- identical(dppm, 0)

  if (noWater)
    NUMVARS <- length(dSet[[2]])
  else NUMVARS <- length(dSet[[2]][-c(water)])
  sSize <- nRows
```

```

# Test if a reference or a test set is generated
if (missing(rSet)) {
  # A reference set is generated
  meanOffSet <- rep(0, NUMVARS)
}
else {
  # A test set is generated
  rSamples <- dim(rSet [[1]]) [1]
  vMethod <- match.arg(vMethod)
  if (identical(vMethod, "mean")){
    # Selection method is MAXMEAN
    ifelse(noWater, rVars <- dSet [[2]],
           rVars <- dSet [[2]][-c(water)])
  }
  else if (identical(vMethod, "stdev")) {
    # Selection method is MAXDEV or MINDEV
    ifelse(noWater, rVars <- dSet [[4]],
           rVars <- dSet [[4]][-c(water)])
  }
  else {
    # Selection method is median
    ifelse(noWater, rVars <- dSet [[5]],
           rVars <- dSet [[5]][-c(water)])
  }
}

# Sort the variables in decreasing or increasing order
varList <- sort(rVars, decreasing = vOrder)[1L:nVars]
varLength <- seq(NUMVARS)
offSetRuns <- seq(varList)
meanOffSet <- numeric(NUMVARS)

for (index in seq(along = varLength)) {
  # Select the variables to mean-shift
  if (noWater) varLabel <- names(dSet [[2]][index])
  else varLabel <- names(dSet [[2]][-c(water)])[index]
  for (vIndex in seq(along = offSetRuns)) {
    offSetLabel <- names(varList)[vIndex]
    nameMatch <- identical(varLabel, offSetLabel)
    if (nameMatch) {
      # if variable satisfies criteria mean-offset
      meanOffSet[index] <- offSet
      break
    }
    else {
      # Do not mean-shift the variable
      meanOffSet[index] <- 0
    }
  }
}
}

# Generate a reference or a test data set
if (noWater)
  tSet <- mvrnorm(nRows, mu = dSet [[2]] +
                 as.numeric(meanOffSet),
                 Sigma = dSet [[3]], empirical = FALSE)

```

```

else tSet <- mvrnorm(nRows, mu = dSet[[2]][-c(water)] +
                    as.numeric(meanOffSet),
                    Sigma = dSet[[3]], empirical = FALSE)
if (identical(nVars, 0)) {
  rownames(tSet) <- as.character(seq(1, nRows))
  if (noWater)
    colnames(tSet) <- colnames(dSet[[1]][[2]])
  else colnames(tSet) <- colnames(dSet[[1]][[2]][-c(water)])
}
else {
  rownames(tSet) <- as.character(seq(sSize + 1,
                                    nRows + sSize))

  if (noWater)
    colnames(tSet) <- colnames(dSet[[1]][[2]])
  else colnames(tSet) <- colnames(dSet[[1]][[2]][-c(water)])
}

# Calculate the mean and standard deviation of the variables in
# the generated data set
tDataMean <- apply(tSet, 2, mean)
tDataStDev <- apply(tSet, 2, sd)

# Construct the required data object
tData <- list(Values = tSet,
             Mean = tDataMean,
             StDev = tDataStDev)

return(tData)
}

```

## C.2.3 Main function: simulateData()

```
#-----#
# AIM                                         #
# Creates the PC scores plots with an LDA boundary or #
# calculates and returns the misclassification rate and the #
# average separation in an experiment.         #
#-----#
# ARGUMENTS                                  #
# dSet:   A data set of class epiData created by #
#         createDataClass().                   #
# rSet:   String corresponding to a generated reference set #
# tSet:   String corresponding to a generated test set for the #
#         same experiment as the rSet.         #
# ofValue: The log value which has been used as an offset for #
#         the generation of the tSet.         #
# rppm:   The number of intervals in which to break the range #
#         of chemical shifts in a spectrum plot. For example, #
#         rpm = 10, means that the chemical shifts from #
#         0 - 10 p.p.m. will be plotted in a spectrum plot. #
# rowScal: A logical value for the row-scaling, here used to #
#         print the appropriate information in the title of #
#         the PCA scores plot.                #
# simTest: Character value to select either to plot the PCA #
#         scores (value "pca") or calculate the statistics #
#         (value "stat").                    #
# runTool: A logical value. If runTool = TRUE then call #
#         plotData(). Default value is FALSE. #
# storePCA: A logical value. If storePCA = TRUE then the object #
#         pcaData is stored in the working environment. #
#         Default value is FALSE.           #
#-----#
# DETAILS                                     #
# -The function simulateData() also contains a call to internal #
# function plotData() which provides additional information #
# for every single run of the simulation algorithm (used for #
# debugging purposes). #
# -simulateData() depends on package *MASS* to perform LDA, #
# using the R function lda(). #
#-----#

simulateData <- function(dSet, rSet, tSet,
                        ofValue, rppm,
                        rowScal = c(TRUE, FALSE),
                        simTest = c("pca", "stat"),
                        runTool = FALSE,
                        storePCA = FALSE)
{
  if (identical(rowScal, TRUE)) {
    # Exponentiate, row-scale to a constant total and
    # log-transform the reference data set
    rSet[[1]] <- exp(rSet[[1]])
    varsRef <- apply(rSet[[1]], 1, sum)
    rSet[[1]] <- rSet[[1]] / varsRef
    rSet[[1]] <- log(rSet[[1]])
  }
}
```

```

# Obtain the mean vector for the new reference set
rSet[[2]] <- apply(rSet[[1]], 2, mean)
# Exponentiate, row-scale to a constant total and
# log-transform the test data set
tSet[[1]] <- exp(tSet[[1]])
varsTest <- apply(tSet[[1]], 1, sum)
tSet[[1]] <- tSet[[1]] / varsTest
tSet[[1]] <- log(tSet[[1]])
# Obtain the new mean vector for the new test set
tSet[[2]] <- apply(tSet[[1]], 2, mean)
}

# Row-bind a reference and a test set
rbSet <- rbind(rSet[[1]], tSet[[1]])
class(rbSet) <- "rBindData"

# Store the row-binded data in the working environment
bdSet <- rbSet

# Perform PCA on the combined data set
weightedMean <- (rSet[[2]] + tSet[[2]]) / 2
pcaData <- prcomp(rbSet, retx = T, center = weightedMean,
                 scale = F, tol = sqrt(.Machine$double.eps))

# Store the PCA data in the working environment
if (identical(storePCA, TRUE)) {
  pcData <- pcaData
}

# Call internal function plotData() for further information
# about the PCA and debugging purposes
if (identical(runTool, TRUE)) plotData(dSet, rbSet, pcaData,
                                       3, rppm)

# Store information about the PCs and their variance in each
# single run of the algorithm.
# Variance of the first PC
VPC1 <- format(pcaData$sdev[1]^2, digits = 2, nsmall = 2)
# Proportion of variance explained by the first PC
PC1 <- format(((pcaData$sdev[1]^2) /
              sum(pcaData$sdev^2)) * 100, digits = 2, nsmall = 2)
# Variance of the second PC
VPC2 <- format(pcaData$sdev[2]^2, digits = 2, nsmall = 2)
# Proportion of variance explained by the second PC
PC2 <- format(((pcaData$sdev[2]^2) /
              sum(pcaData$sdev^2)) * 100, digits = 2, nsmall = 2)
# Number of PCs to extract
RPCS <- length(pcaData$sdev)
pc <- c(VPC1, PC1, VPC2, PC2, RPCS)

# Create a vector of type factor, with colouring information
# for the points of the two data sets in the PCA scores plots
isColour <- character(dim(rbSet)[1])
isArtificial <- numeric(dim(rbSet)[1])
rSize <- dim(rSet[[1]])[1]

```

```

for (i in seq(along = rownames(rbSet))) {
  isArtificial[i] <- (as.numeric(rownames(rbSet)[[i]]) >=
                      rSize + 1)
  ifelse(isArtificial[i], isColour[i] <- "#00A5C6",
         isColour[i] <- "#AD4A18")
}

# Calculate the average separation
totSamples <- dim(rbSet)[1]
tSize <- totSamples - rSize
distData <- as.matrix(dist(pcaData$x[, 1:2],
                          method = "euclidean"))
samplesDistance <- sum(distData[1:rSize,
                              (rSize + 1):totSamples])
SamplesNumProduct <- tSize * (totSamples - tSize)
averSep <- samplesDistance / SamplesNumProduct

# Perform LDA on the first two PCs
ldaData <- lda(pcaData$x[, 1:2], grouping = isArtificial,
              method = "moment")

if (identical(simTest, "pca")) {
  # Plot PCA scores
  if (rowScal) title <- "ROW SCALED Data"
  else title <- "UNSCALED Data"
  plotTitle <- paste(title, " - Offset: ",
                    format(exp(as.numeric(ofValue)),
                            digits = 2, nsmall = 2), "\n",
                    "Sample Size: ", rSize, sep = "")
  # Calculate the slope of the LDA boundary
  slope <- -(ldaData$scaling[1, 1] / ldaData$scaling[2, 1])
  # Adjust the plotting settings for the scores plot
  par(mar=c(4, 4, 4, 2) + 0.1, mgp=c(2, 0.5, 0), tcl = -0.5,
      xpd = FALSE, xaxs = "i", yaxs = "i")
  # Plot the PCA scores for the first two PCs
  plot(pcaData$x[, 1:2], pch = ".", col = isColour,
       cex.main = 0.5, cex.lab = 0.5, cex.axis = 0.5,
       main = plotTitle)
  # Plot the centre of the points in the reference set
  points(ldaData$means[1, 1], ldaData$means[1, 2],
        col = "#632910", pch = 8, cex = 2.0)
  # Plot the centre of the points in the test set
  points(ldaData$means[2, 1], ldaData$means[2, 2],
        col = "#00394A", pch = 8, cex = 2.0)
  # Plot the LDA boundary
  abline(a = 0, b = slope, col = "#8C007B", lwd = 1.5)
  # Add the Average Separation value to the scores plot
  textSide <- 3
  mtext(paste("Average Separation: ",
             format(averSep, digits = 4), sep = ""),
        adj = 0, side = textSide, cex = 0.3)
}
else {
  # Obtain and process the information from the LDA, about the
  # misclassified samples in the two data sets
  predSet <- predict(ldaData)
}

```

```

classType <- table(predSet [[1]][1:rSize])
isRefTrue <- (identical(classType [[1]], 100L) &&
             identical(classType [[2]], 0L))
isTestTrue <- (identical(classType [[1]], 0L) &&
              identical(classType [[2]], 100L))
isCorrect <- isRefTrue || isTestTrue
if (isCorrect) {
  rError <- tError <- 0
  if (identical(classType [[2]], 0L)) {
    rGroup <- 0
    tGroup <- 1
  }
  else {
    rGroup <- 1
    tGroup <- 0
  }
}
else {
  rClass <- predSet [[1]][1:rSize]
  tClass <- predSet [[1]][(rSize + 1):totSamples]
  groupOneSize <- table(predSet [[1]][1:rSize]) [[1]]
  groupTwoSize <- rSize - groupOneSize
  if (groupOneSize > groupTwoSize) {
    rGroup <- 0
    tGroup <- 1
    rError <- table(rClass) [[2]]
    tError <- table(tClass) [[1]]
  }
  else {
    rGroup <- 1
    tGroup <- 0
    rError <- table(rClass) [[1]]
    tError <- table(tClass) [[2]]
  }
}
}

# Calculate the LDA misclassification rate
ldaError <- ((rError + tError) / totSamples) * 100

# Construct the object containing the misclassification
# rate, the average separation and the PCs information
# in each experiment
predSet$classes <- list(ref = rGroup, test = tGroup)
predSet$aversep <- averSep
predSet$error <- ldaError
predSet$pc <- pc
}

# If a PCA scores plot is not required, return the statistics
# information
isStat <- identical(simTest, "stat")
ifelse(isStat, return(predSet),
       return(list(ldaData, averSep, pc)))
}

```



## C.2.4 Main function: runSimulation()

```
#-----#
# AIM                                         #
# The main function, which runs the simulation algorithm. #
#-----#
# ARGUMENTS                                  #
# dSet:   A data set of class epiData created by      #
#         createDataClass().                        #
# water:  A vector giving the range of water variables to #
#         exclude from the data e.g. 153:158 for the spectral #
#         data with 0.04 bin width.                #
# sValue: A real value corresponding to the first value of #
#         an offset range of values for an experiment.    #
# eValue: A real value corresponding to the last value of #
#         an offset range of values for an experiment.    #
# stepSize: A real value corresponding to the step size between #
#         offsets in the selected offset range.          #
# nRuns:   The number of runs of an experiment.        #
# nPars:   The number of parameters (statistics) to store #
#         information about. In the standard case, nPars is #
#         equal to 2, as we have two statistics, the #
#         misclassification rate and the average separation #
#         for which we store information.              #
# nVars:   The number of variables to mean-shift. The number is #
#         set to 0 when a reference set is generated.    #
# rSize:   The number of samples in the reference set.   #
# tSize:   The number of samples in the test set.       #
# vMethod: Character string corresponding to the method to #
#         select the variables for mean-shifting. Three #
#         methods are available: stdev, mean and median. #
# vOrder:  A logical value. If TRUE, the variables are #
#         selected in decreasing order, otherwise in #
#         increasing order.                            #
# rppm:    The number of intervals in which to break the range #
#         of chemical shifts in a spectrum plot. For example, #
#         rpm = 10, means that the chemical shifts from #
#         0 - 10 p.p.m. will be plotted in a spectrum plot. #
# dppm:    The number of downfield variables to exclude from #
#         the data e.g. dppm = 26 means, that the first 25 #
#         variables (corresponding to the variables with #
#         chemical shifts above 9.98 p.p.m.) will be excluded #
#         from the analyses.                          #
# rowScal: A logical value for the row-scaling.         #
# plotTrue: A logical value. If plotTrue = TRUE then the PCA #
#         scores are plotted and if FALSE, the statistics #
#         are calculated. Default value is FALSE.      #
# runTool:  A logical value. If runTool = TRUE then call #
#         plotData(). Default value is FALSE.         #
# storePCA: A logical value. If storePCA = TRUE then the object #
#         pcaData is stored in the working environment. #
#         Default value is FALSE.                    #
# multiple: A logical value. If multiple = TRUE then #
#         plotRows * plotCols scores plots are plotted. #
#         Default value is FALSE.                    #
```

```

# plotRows:The number of scores plots in a column of a multiple #
# plot.                                                                #
# plotCols:The number of scores plots in a row of a multiple         #
# plot.                                                                #
#-----#
# DETAILS                                                                #
# -If sValue, eValue and stepSize are e.g. 0.4, 0.6 and 0.1         #
# respectively, then the experiment will be executed for             #
# offsets 0.4, 0.5 AND 0.6.                                          #
# -Starting a graphics device driver for X requires a machine       #
# with access to an X server. A different graphics device          #
# driver may be needed in a Windows (e.g. cairo) or Mac OS X       #
# System (quartz).                                                  #
# -runSimulation() depends on package *xtable* to convert the      #
# statistics information matrix to a latex table, using R           #
# function xtable().                                                #
#-----#

runSimulation <- function(dSet, water, sValue, eValue, stepSize,
                          nRuns, nPars, nVars, rSize, tSize,
                          vMethod, vOrder, rppm, dppm,
                          rowScal = c(TRUE,FALSE),
                          plotTrue = FALSE, runTool = FALSE,
                          storePCA = FALSE, multiple = FALSE,
                          plotRows = NULL, plotCols = NULL)
{
  # Calculate the offset range
  startValue <- sValue
  endValue <- eValue
  ofValue <- seq(startValue, endValue, by = stepSize)
  nCols <- length(ofValue)

  # Set some bounds for counters
  seqPlots <- seq(1, nCols, by = 1)
  seqRuns <- seq(1, nRuns, by = 1)

  # Create a list to store the statistics information
  setsInfo <- list(misrate = matrix(nrow = nRuns, ncol = nCols),
                  aversep = matrix(nrow = nRuns, ncol = nCols))

  if (plotTrue) {
    # Plot the PCA scores plot
    if (multiple) {
      # Start a graphics device driver for X with width and
      # height of the plotting window with respect to plotCols
      # and plotRows
      x11(width = plotCols * 3, height = plotRows * 3)
      par(mfrow = c(plotRows, plotCols))
    }
    simTest <- "pca"
  }
  else {
    # Calculate the statistics
    simTest <- "stat"
  }
}

```

```

# If a call to plotData() is required, start a graphics
# device driver for X with width and height equal to 8 inches
if (identical(runTool, TRUE)) x11(width = 8, height = 8)

# Execute the simulation algorithm nRuns times
for (run in seq(along = seqRuns)) {
  print(paste("Run: ", run, sep = ""))
  for (index in seq(along = seqPlots)) {
    # Generate a reference set
    refSet <- generateSet(dSet, water, dppm, 0, 0, rSize)
    # Generate a test set
    testSet <- generateSet(dSet, water, dppm, ofValue[index],
                          nVars, tSize, vMethod, vOrder,
                          refSet)
    # Plot PCA scores or obtain the statistics
    ldaInfo <- simulateData(dSet, refSet, testSet,
                           as.character(ofValue[index]),
                           rppm, rowScal, simTest,
                           runTool, storePCA)

    if (identical(simTest, "stat")) {
      # If statistics have been obtained, store them
      # for each single run, to the created list
      ldaError <- format(ldaInfo$error, digits = 2,
                         nsmall = 1)
      setsInfo$misrate[run, index] <- as.numeric(ldaError)
      ldaAverSep <- format(ldaInfo$aversep, digits = 4,
                          nsmall = 4)
      setsInfo$aversep[run, index] <- as.numeric(ldaAverSep)
    }

    if (identical(simTest, "pca")) {
      # Send to the standard output information about the
      # offset
      print(paste("      Offset (log): ",
                  format(ofValue[index], digits = 2,
                         nsmall = 2),
                  "      Offset: ",
                  format(exp(ofValue[index]),
                         digits = 2, nsmall = 2),
                  sep = ""))
    }
    else {
      # Send to the standard output information about the
      # offset, the PCs and their variance for each single
      # experiment
      print(paste("      Offset (log): ",
                  format(ofValue[index], digits = 2,
                         nsmall = 2),
                  "      Offset: ",
                  format(exp(ofValue[index]),
                         digits = 2, nsmall = 2),
                  sep = ""))
      print(paste("      VPC1: ", ldaInfo$pc[1],
                  "      PC1: ", ldaInfo$pc[2],
                  "      VPC2: ", ldaInfo$pc[3],

```

```

        "      PC2: ", ldaInfo$pc[4],
        "      RPCS: ", ldaInfo$pc[5], sep = "")
    }
}

if (identical(simTest, "stat")) {
  # If statistics have been obtained, create a matrix to store
  # the mean values of the statistics in nRuns of the algorithm
  setSeparation <- matrix(nrow = nPars, ncol = nCols)
  rownames(setSeparation) <- c("Error Rate (%)",
                              "Average Separation")
  colnames(setSeparation) <- format(exp(ofValue), digits = 2,
                                    nsmall = 2)

  # Obtain the mean values for each statistic and store them to
  # the created matrix
  seqPars <- seq(1, nPars, by = 1)
  for (param in seq(along = seqPars)) {
    for (index in seq(along = seqPlots)) {
      setSeparation[param, index] <-
        mean(setsInfo[[param]][1:nRuns, index])
    }
  }

  # Convert the R information in the matrix to a latex table
  align <- paste("|", paste(rep("c", nCols), collapse = ""),
                sep = "")
  setParams <- xtable(setSeparation, align = align,
                    digits = 4)

  class(setSeparation) <- "statData"
  return(list(setSeparation, setsInfo, setParams))
}
}

```

## C.2.5 Main function: plotMeanShifting()

```
#-----#
# AIM #
# Plots the mean shifting results in each simulation experiment.#
#-----#
# ARGUMENTS #
# dSet: A data set of class epiData created by #
# createDataClass(). #
# water: A vector giving the range of water variables to #
# exclude from the data e.g. 153:158 for the spectral #
# data with 0.04 bin width. #
# smpSize: The number of samples in the reference and the test #
# set. #
# offSet: A log value to add to the means of the selected #
# variables for mean-shifting, in the test set. To #
# generate a reference data set, an offset of size 0 #
# is used. #
# nVars: The number of variables to mean-shift. The number is #
# set to 0 when a reference set is generated. #
# vMethod: Character string corresponding to the method to #
# select the variables for mean-shifting. Three #
# methods are available: stdev, mean and median. #
# vOrder: A logical value. If TRUE, the variables are #
# selected in decreasing order, otherwise in #
# increasing order. #
# rppm: The number of intervals in which to break the range #
# of chemical shifts in a spectrum plot. For example, #
# rpm = 10, means that the chemical shifts from #
# 0 - 10 p.p.m. will be plotted in a spectrum plot. #
#-----#
# DETAILS #
# -plotMeanShifting() uses the R function plotMeanSpectrum() to #
# plot the two mean spectra in each experiment. #
# -plotMeanShifting() calls R function generateSet() to #
# generate reference and test sets for comparison of their #
# means in each simulation experiment. #
#-----#
```

```
plotMeanShifting <- function(dSet, water, smpSize, offSet,
                             nVars, vMethod, vOrder, rppm)
{
  # Start a graphics device driver for X with width 7 inches
  # and height 3 inches, and adjust specific plotting settings
  x11(width = 7, height = 3)
  par(mar = c(4, 4, 2, 2) + 0.1, mgp = c(2, 0.5, 0), tcl = -0.5,
      xpd = FALSE, yaxs = "i")

  # Generate a reference set
  rSet <- generateSet(dSet, water, 0, 0, 0, smpSize, vMethod,
                    vOrder)

  # Generate a test set
  tSet <- generateSet(dSet, water, 0, offSet, nVars, smpSize,
                    vMethod, vOrder, rSet)
```

```

# Adjust the water variables in the generated data sets
rSet[[2]] <- append(rSet[[2]], c(NA,NA,NA,NA,NA,NA), after = 127)
names(rSet[[2]])[water] <- c("4.90", "4.86", "4.82",
                             "4.78", "4.74", "4.70")

tSet[[2]] <- append(tSet[[2]], c(NA,NA,NA,NA,NA,NA), after = 127)
names(tSet[[2]])[water] <- c("4.90", "4.86", "4.82",
                             "4.78", "4.74", "4.70")

# Calculate the y-limits for the mean-shifting plot
plotMin <- min(rSet[[2]][-c(water)], tSet[[2]][-c(water)])
plotMax <- max(rSet[[2]][-c(water)], tSet[[2]][-c(water)])
yLim <- c(plotMin, plotMax)

# Set the title for the mean spectra plot
pTitle <- NULL

# Create the mean-shifting plot for the two data sets.
plotMeanSpectrum(rSet[[2]], rppm, pTitle, "#AD4A18", yLim)

# Add the mean spectrum of the test set to the previous plot
lines(tSet[[2]], col = "#00A5C6", lwd = 1.1)
}

```

## C.2.6 Main function: plotBoundaries()

```
#-----#
# AIM                                         #
# Creates the Figures with the three PCA scores plots in each #
# experiment.                               #
#-----#
# ARGUMENTS                                  #
# dSet:   A data set of class epiData created by           #
#         createDataClass().                             #
# water:  A vector giving the range of water variables to  #
#         exclude from the data e.g. 153:158 for the spectral #
#         data with 0.04 bin width.                     #
# v1, v2, #
# v3:     Three offsets for the three PCA scores plots,   #
#         corresponding to misclassification rates of 20, 10 #
#         and 1 % respectively.                          #
# nVars:  The number of variables to mean-shift. The number is #
#         set to 0 when a reference set is generated.     #
# sSize:  The number of samples in the reference and the test #
#         set.                                           #
# vMethod: Character string corresponding to the method to #
#         select the variables for mean-shifting. Three   #
#         methods are available: stdev, mean and median.  #
# vOrder: A logical value. If TRUE, the variables are    #
#         selected in decreasing order, otherwise in     #
#         increasing order.                              #
# rppm:   The number of intervals in which to break the range #
#         of chemical shifts in a spectrum plot. For example, #
#         rpm = 10, means that the chemical shifts from   #
#         0 - 10 p.p.m. will be plotted in a spectrum plot. #
# dppm:   The number of downfield variables to exclude from #
#         the data e.g. dppm = 26 means, that the first 25 #
#         variables (corresponding to the variables with   #
#         chemical shifts above 9.98 p.p.m.) will be excluded #
#         from the analyses.                             #
# rowScal: A logical value for the row-scaling.          #
#-----#

plotBoundaries <- function(dSet, water, v1, v2, v3, nVars,
                           sSize, vMethod, vOrder, rppm,
                           dppm, rowScal = FALSE)
{
  # Start a graphics device driver for X with width 7 inches
  # and height 2.5 inches, and adjust specific plotting settings
  x11(width = 7, height = 2.5)
  par(mar = c(4, 4, 4, 2) + 0.1, mgp = c(2, 0.5, 0), tcl = -0.5,
      xpd = FALSE, xaxs = "i", yaxs = "i")
  par(mfrow = c(1, 3))

  # Create a list of offsets
  vList <- c(v1, v2, v3)
```

```
# Plot the PCA scores for the three offsets
for (i in 1:length(vList)) {
  # Plot PCA scores of for offset i
  runSimulation(dSet, water, vList[i], vList[i], 1, 1, 2,
               nVars, sSize, sSize, vMethod, vOrder, rppm,
               dppm, rowScal, TRUE, FALSE, FALSE, FALSE)
}
```



## C.2.7 Main function: plotSimStats()

```
#-----#
# AIM #
# Creates the comparison plots of the two statistics versus #
# the offset values for an experiment (misclassification rate #
# versus offset, etc). #
#-----#
# ARGUMENTS #
# parData: A data set of class statData created by #
# runSimulation(). #
# sValue: A real value corresponding to the first value of #
# an offset range of values for an experiment. #
# eValue: A real value corresponding to the last value of #
# an offset range of values for an experiment. #
# stepSize: A real value corresponding to the step size between #
# offsets in the selected offset range. #
#-----#
# DETAILS #
# -To obtain a parData object, a user must store the output of #
# runSimulation() to an object #
# e.g. parData <- runSimulation(...). #
# -plotSimStats() depends on the package *Hmisc* to plot the #
# error bars in the comparison plots. #
# -plotSimStats() uses the R function createStatsPlots() to #
# create the plots for the two statistics. #
#-----#

plotSimStats <- function(parData, sValue, eValue, stepSize)
{
  # Calculate the offset range
  ofValue <- seq(sValue, eValue, by = stepSize)

  # Start a graphics device driver for X with width 9 inches
  # and height 3 inches, and adjust specific plotting settings
  x11(width = 9, height = 3)
  par(mar=c(4, 4, 4, 2) + 0.1, tcl = -0.5,
      xpd = FALSE)
  par(mfrow = c(1,3))

  # Calculate the maximum error
  maxErr <- max(parData [[2]][[1]])

  # Set the x-axis limits for the error vs offsets plot
  xLim <- c(exp(sValue), exp(eValue))

  # Set the y-axis limits for the error vs offsets plot
  yLim <- c(0, maxErr)

  # The title of the error vs offsets plot
  pTitle <- "Error Rate Vs Offset"

  # Plot the misclassification rate versus offsets
  createStatsPlots(ofValue, parData, pTitle, "#00A5C6",
                  xLim, yLim, 1)
}
```

```

# Set the x-axis limits for the average separation
# versus offsets plot
xLim <- c(exp(sValue), exp(eValue))

# Set the y-axis limits for the average separation
# versus offsets plot
minASep <- min(parData[[2]][[2]])
maxASep <- max(parData[[2]][[2]])
yLim <- c(minASep, maxASep)

# The title of the error vs offsets plot
pTitle <- "Average Separation Vs Offset"

# Plot the misclassification rate versus offsets
createStatsPlots(ofValue, parData, pTitle, "#00A5C6",
                 xLim, yLim, 2)

# Set the x-axis limits for the error rate versus
# average separation plot
xLim <- c(minASep, maxASep)

# Set the y-axis limits for the error rate versus
# average separation plot
yLim <- c(0, maxErr)

# The title of the error vs offsets plot
pTitle <- "Error Rate Vs Average Separation"

# Plot the misclassification rate versus offsets
createStatsPlots(ofValue, parData, pTitle, "#00A5C6",
                 xLim, yLim, 3)
}

```

## C.2.8 Main function: computeCV()

```
#-----#
# AIM                                         #
# Computes the coefficient of variation (CV) for the #
# misclassification rate or the average separation in a #
# simulation 100-run experiment.             #
#-----#
# ARGUMENTS                                  #
# parData: A data set of class statData created by #
#           runSimulation().                   #
# sOption: A character string to select the statistic to #
#           compute the CV for. If "error", then the CV of the #
#           misclassification rate is computed, otherwise if it #
#           is "aversep", the average separation and if "both" #
#           the CV for both statistics is computed.         #
#-----#

computeCV <- function(parData,
                      sOption = c("error", "aversep", "both"))
{
  # Define number of columns
  cvCols <- ncol(parData[[2]][[1]])

  # Define row names for the output
  rNames <- c("Error Rate (CV)", "Error Rate (StDev)",
             "Error Rate (Mean)", "Average Separation (CV)",
             "Average Separation (StDev)",
             "Average Separation (Mean)")

  # Set options depending on the type of variable
  if (identical(sOption, "error")) {
    rLabs <- rNames[1:3]
    nRow <- 3
    pData <- parData[[2]][[1]]
  }
  else if (identical(sOption, "aversep")) {
    rLabs <- rNames[4:6]
    nRow <- 3
    pData <- parData[[2]][[2]]
  }
  else {
    rLabs <- rNames
    nRow <- 6
    pData <- parData[[2]]
  }

  # Define dimnames for the CV matrix
  cLabs <- attr(parData[[1]], "dimnames")[[2]]

  # Create the CV matrix
  cvMat <- matrix(NA, nrow = nRow, ncol = cvCols, byrow = TRUE,
                 dimnames = list(rLabs, cLabs))
}
```

```

# Compute the CV, stdev and mean of a variable
cvCalc <- function(pData, nRow, cvCols) {

  # Create a matrix to store the CV information
  cvData <- matrix(NA, nrow = nRow,
                  ncol = cvCols, byrow = TRUE)

  # Calculates the CV of the variable
  funCV <- function(x) (sd(x) / mean(x)) * 100

  # Stores the CV, StDev and mean information to
  # the cvData matrix for further use
  cvData[1, ] <- apply(pData[, 1:cvCols], 2,
                     funCV)

  cvData[2, ] <- apply(pData[, 1:cvCols], 2, sd)
  cvData[3, ] <- apply(pData[, 1:cvCols], 2,
                     mean)

  #Return the CV information
  return(cvData)
}

# Store the computed information to the CV matrix
if (identical(sOption, "error") ||
    identical(sOption, "aversep"))
  cvMat[1:3, ] <- cvCalc(pData, nRow, cvCols)
else {
  nRow <- 3
  cvMat[1:3, ] <- cvCalc(pData[[1]], nRow, cvCols)
  cvMat[4:6, ] <- cvCalc(pData[[2]], nRow, cvCols)
}

# Return the CV matrix to the standard output
return(round(cvMat, digits = 2))
}

```

## C.2.9 Internal function: plotData()

```
#-----#
# AIM #
# Provides additional information for every single run of the #
# simulation algorithm. It is to be used for debugging #
# purposes only. #
#-----#
# ARGUMENTS #
# dSet: A data set of class epiData created by #
# createDataClass(). #
# bdSet: An object of class rBindData created by function #
# simulateData(). #
# pcData: An object of class prcomp created by function #
# simulateData(). #
# nPC: The number of principal components to plot #
# information about. #
# rppm: The number of intervals in which to break the range #
# of chemical shifts in a spectrum plot. For example, #
# rpm = 10, means that the chemical shifts from #
# 0 - 10 p.p.m. will be plotted in a spectrum plot. #
#-----#
# DETAILS #
# -plotData() calls the internal functions, plotMeansSpectrum(), #
# plotScores(), plotLoads() and plotVarExpl() to plot #
# information about the PCs loadings and the variance #
# explained by the PCs. In addition, it plots the mean and #
# stdev spectra for the reference and the test set in an #
# experiment, as well as the standard deviation of all PCs #
# and the scores plots for the first three PCs. #
#-----#

plotData <- function(dSet, bdSet, pcData, nPC, rppm)
{
  # Adjust the layout settings for the required plots
  par(mar = c(4, 5, 3, 3) + 0.1, tcl = -0.5, xpd = FALSE)
  layout(rbind(c(1, 1, 1, 3),
               c(2, 2, 2, 6),
               c(4, 4, 4, 7),
               c(5, 5, 8, 9)))

  # Extract the reference set information and statistics
  # (mean, stdev) from the row-binded set
  ref <- bdSet[1 : (dim(bdSet)[[1]] / 2), ]
  refMean <- apply(ref, 2, mean)
  refDev <- apply(ref, 2, sd)

  # Extract the test set information and statistics
  # (mean, stdev) from the row-binded set
  test <- bdSet[((dim(bdSet)[[1]] / 2) + 1) : nrow(bdSet), ]
  testMean <- apply(test, 2, mean)
  testDev <- apply(test, 2, sd)
}
```

```

# Calculate the y-axis limits of the mean spectra plot
plotMin <- min(refMean, testMean)
plotMax <- max(refMean, testMean)
yLim <- c(plotMin, plotMax)

# Set the title of the mean spectra plot
pTitle <- "Mean spectra"

# Plot the mean spectra of the reference and the test set
plotMeanSpectrum(refMean, rppm, pTitle, "#AD4A18", yLim)

# Plot the mean spectra for the test set
lines(testMean, col = "#00A5C6", lwd = 1.1)

# Calculate the y-axis limits of the stdev spectra plot
plotMin <- min(refDev, testDev)
plotMax <- max(refDev, testDev)
yLim <- c(plotMin, plotMax)

# Set the title of the stdev spectra plot
pTitle <- "Stdev spectra"

# Plot the standard deviation spectra of the reference and
# the test set
plotMeanSpectrum(refDev, rppm, pTitle, "#AD4A18", yLim)

# Plot the stdev spectra for the test set
lines(testDev, col = "#00A5C6", lwd = 1.1)

# plot the standard deviation of the PCs
plot(pcData$sdev,
     main = "Standard Deviation of PCs",
     cex.main = 0.6, cex.lab = 0.5, cex.axis = 0.5,
     xlab = "Component",
     ylab = "Standard Deviation",
     xlim = c(0, length(pcData$sdev)),
     pch = ".", cex = 3.0)

# Plot the PCs loadings versus the variables
plotLoads(bdSet, pcData, nPC, rppm)

# Plot the variance explained by the PCs
plotVarExpl(pcData)

# Plot a biplot of the first two PCs
biplot(pcData, main = "Biplot of first 2 PCs",
       cex.main = 0.6, cex.lab = 0.5, cex.axis = 0.5)

# Set the title of the scores plot for PC1 and PC2
pTitle <- "Scores of PC1 and PC2"

# Plot the PCs scores for the first two PCs
plotScores(dSet, pTitle, "red", 1, 2, pcData)

# Set the title of the scores plot for PC1 and PC3
pTitle <- "Scores of PC1 and PC3"

```

```
# Plot the PCs scores for the first and the third PC
plotScores(dSet, pTitle, "green", 1, 3, pcData)

# Set the title of the scores plot for PC2 and PC3
pTitle <- "Scores of PC2 and PC3"

# Plot the PCs scores for the second and the third PC
plotScores(dSet, pTitle, "blue", 2, 3, pcData)
}
```

## C.2.10 Internal function: plotVarExpl()

```
#-----#
# AIM                                         #
# Creates a barplot of the proportion of total variation #
# explained by the first ten PCs.           #
#-----#
# ARGUMENTS                                  #
# pcData: An object of class pcaInfo created by function #
#         simulateData().                       #
#-----#
# DETAILS                                    #
# -plotVarExpl() can be used with pcaInfo objects obtained by #
# either princomp() or prcomp().             #
#-----#

plotVarExpl <- function(pcData)
{
  # Set the colours for the boxes of the standard deviations of
  # the first ten PCs
  colorset <- c("#BDC6DE", "#949CCE", "#6373B5", "#3152A5",
               "#083194", "#082984", "#08296B", "#08215A",
               "#00184A", "#180042")

  # Calculate the percentage of variation explained by each PC
  pcs <- numeric(10)
  for (p in (1:10)) {
    pcs[p] <- as.numeric(format(((pcData$sdev[p]^2) /
                                sum(pcData$sdev^2)) * 100,
                                digits = 2, nsmall = 2))
  }

  # Stores the scores information, checking which PCA function
  # has been used
  if (identical(pcData$scores, NULL)) scores <- pcData$x
  else scores <- pcData$scores

  names(pcs) <- colnames(scores)[1:10]

  # Plot the bar plot with the PCs
  pcabar <- barplot(pcs, names.arg = names(pcs),
                   beside = TRUE, col = colorset, border = NA,
                   main = "",
                   xlab = "Component",
                   ylab = "Proportion of Total Variation
                           Explained (%)",
                   ylim = c(0, 110), xpd = TRUE,
                   axes = TRUE, axisnames = TRUE,
                   cex.axis = 0.7, cex.names = 0.7,
                   cex.lab = 0.7, cex.main = 0.8)
  text(pcabar, pcs, labels = pcs, col = rev(colorset), pos = 3,
       cex = 0.7)
}
```



## C.2.11 Internal function: plotLoads()

```
#-----#
# AIM #
# Plots the PCs loadings versus the variables. #
#-----#
# ARGUMENTS #
# bdSet: An object of class rBindData created by function #
# simulateData(). #
# pcData: An object of class pcaInfo created by function #
# simulateData(). #
# nPC: The number of principal components to plot #
# information about. #
# rppm: The number of intervals in which to break the range #
# of chemical shifts in a spectrum plot. For example, #
# rpm = 10, means that the chemical shifts from #
# 0 - 10 p.p.m. will be plotted in a spectrum plot. #
#-----#

plotLoads <- function(bdSet, pcData, nPC, rppm)
{
  # Store the minimum and maximum loading for the first nPC PCs
  minRot <- round(min(pcData$rotation[ , 1:nPC]), digits = 3)
  maxRot <- round(max(pcData$rotation[ , 1:nPC]), digits = 3)

  # Adjust the layout settings for the required plots
  par(mar = c(5, 5, 1, 1) + 0.1)

  # Plot the loadings for the first PC
  plot(pcData$rotation[ , 1],
       type = "n",
       xlab = "Chemical Shift (ppm)",
       ylab = "Loadings",
       col = rainbow(nPC)[1],
       ylim = c(minRot, maxRot),
       xaxt = "n",
       yaxt = "n")
  dfield <- dim(bdSet)[2]
  ppmInterval <- dfield / rppm
  axis(1, at = seq(0, dfield, ppmInterval),
       labels = seq(rppm, 0, -1), cex = 0.6)
  axis(2, at = seq(minRot, maxRot, 0.1), cex = 0.6)

  # Plot the loadings for the next nPC-1 PCs
  k <- 0
  sComp <- seq(1L, nPC, by = 1L)
  for (pc in seq(along = sComp)) {
    # Adds the pcth component's loadings line to the existing
    # loadings plot
    k <- k + 0.1
    lines(pcData$rotation[ , pc], type = "|",
         col = rainbow(nPC)[pc], lwd = 1.1)
  }
}
```

```

#Add a legend to the plot
pcLoads <- abs(minRot) >= abs(maxRot)
location <- ifelse(pcLoads, "bottomright", "topright")
legendText <- c(paste("PC", as.character(1:nPC), sep=" "))
legend(location,
        legend = legendText,
        col = rainbow(nPC),
        lty = 1,
        bty = "n",
        xjust = 0,
        cex = 0.8)
abline(h = 0, lwd = 1.2, lty = 2)
}

```

## C.2.12 Internal function: createStatsPlots()

```
#-----#
# AIM                                         #
# Plots the comparisons of the two statistics versus offsets. #
#-----#
# ARGUMENTS                                  #
# ofValue: The log value which has been used as an offset for #
#           the generation of the tSet.           #
# parData: A data set of class statData created by           #
#           runSimulation().                               #
# pTitle: Character string to give the title of the plot.    #
# sCol:    The colour of the line of a statistics' values.  #
# xLim:    The limits of the values in the x-axis of the plot. #
# yLim:    The limits of the values in the y-axis of the plot. #
# tPlot:   A numeric value (1,2 or 3) for the type of plot.  #
#-----#
# DETAILS                                    #
# -createStatsPlots() depends on package *Hmisc* to plot the #
# vertical error bars in the comparison plots, using R       #
# function errbar().                                         #
#-----#
```

```
createStatsPlots <- function(ofValue, parData, pTitle, sCol,
                             xLim, yLim, tPlot)
{
  # Set the type of plot required
  if (identical(tPlot, 1)) {
    # Plot of type error vs offset
    stat1 <- exp(ofValue)
    stat2 <- parData[[1]][1, ]
    pDat <- parData[[2]][[1]]
    xLab <- "Offset"
    yLab <- "Error Rate (%)"
  }
  else if (identical(tPlot, 2)) {
    # Plot of type aversep vs offset
    stat1 <- exp(ofValue)
    stat2 <- parData[[1]][2, ]
    pDat <- parData[[2]][[2]]
    xLab <- "Offset"
    yLab <- "Average Separation"
  }
  else {
    # Plot of type error vs aversep
    stat1 <- parData[[1]][2, ]
    stat2 <- parData[[1]][1, ]
    pDat <- parData[[2]][[1]]
    xLab <- "Average Separation"
    yLab <- "Error Rate (%)"
  }
}
```

```
# Plot the misclassification rate versus offsets
plot(stat1, stat2, type = "b", main = pTitle, col = sCol,
      xlab = xLab, ylab = yLab, xlim = xLim, ylim = yLim)
minStat <- apply(pDat, 2, min)
maxStat <- apply(pDat, 2, max)
errbar(stat1, stat2, maxStat, minStat, add = TRUE)
}
```

# Appendix D

---

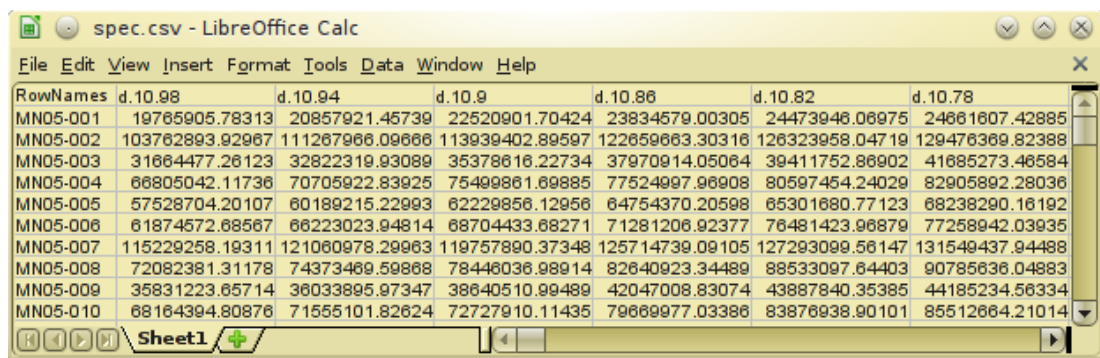
## User Guide for the Simulation Algorithm

### Contents

1. Data Input and Pre-treatment - `createDataClass()`
2. Reference and Test Sets - `generateSet()`
3. Simulation Analyses - `simulateData()`
4. Execution of the Simulation - `runSimulation()`
5. The Effect of Mean-Shifting - `plotMeanShifting()`
6. Plot PCs Scores and LDA Boundary - `plotBoundaries()`
7. Plot Statistics vs Offsets - `plotSimStats()`
8. Plot Additional Information - `plotData()`

## D.1 Data Input and Pre-treatment

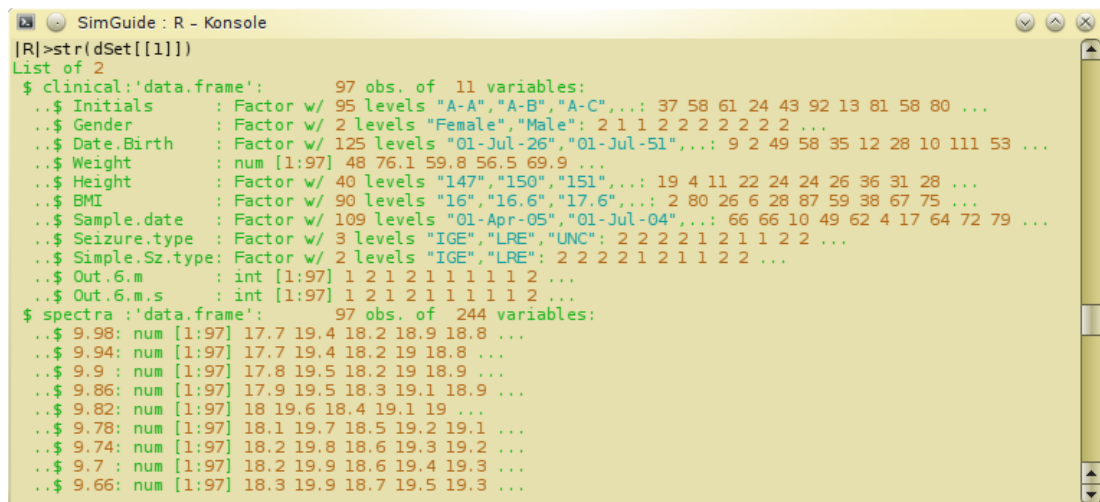
The original epilepsy data consists of two data files in the Comma Separated Value (.csv) file format. This is a file format commonly used to exchange data between various applications, and its usual form is that used in spreadsheets. The first .csv file contains the clinical characteristics information of the patients, as described in Chapter 2. The second .csv file contains the spectral information of the patients. More specifically, the intensity levels of the metabolites of the blood serum of the patients, obtained by  $^1\text{H}$  NMR spectroscopy. A formatted part of such a file can be seen in Figure D.1 for 0.04 ppm bin width data. Function



RowNames	d.10.98	d.10.94	d.10.9	d.10.86	d.10.82	d.10.78
MN05-001	19765905.78313	20857921.45739	22520901.70424	23834579.00305	24473946.08975	24861807.42885
MN05-002	103762893.92967	111267966.09666	113939402.89597	122659663.30316	126323958.04719	129476369.82388
MN05-003	31864477.26123	32822319.93089	35378616.22734	37970914.05064	39411752.86902	41885273.46584
MN05-004	66805042.11736	70705922.83925	75499861.69885	77524997.96908	80597454.24029	82905892.28036
MN05-005	57528704.20107	60189215.22993	62229856.12956	64754370.20598	65301680.77123	68238290.16192
MN05-006	61874572.68567	66223023.94814	68704433.68271	71281206.92377	76481423.96879	77258942.03935
MN05-007	115229258.19311	121060978.29963	119757890.37348	125714739.09105	127293099.56147	131549437.94488
MN05-008	72082381.31178	74373469.59868	78446036.98914	82640923.34489	88533097.64403	90785636.04883
MN05-009	35831223.65714	36033895.97347	38640510.99489	42047008.83074	43887840.35385	44185234.56334
MN05-010	68164394.80876	71555101.82624	72727910.11435	79669977.03386	83876938.90101	85512664.21014

Figure D.1: Example of a .csv file containing the spectral information of the epilepsy patients.

createDataClass() is responsible for inputting these two data files to R as data frames, joining and storing the information to a data list of class epiData for further use in the analyses. Figure D.2 shows an example of such a list with the clinical and spectral information. The epiData list stores also the mean,



```
[R]>str(dSet[[1]])
List of 2
 $ clinical:'data.frame':
  97 obs. of 11 variables:
  ..$ Initials      : Factor w/ 95 levels "A-A","A-B","A-C",...: 37 58 61 24 43 92 13 81 58 80 ...
  ..$ Gender        : Factor w/ 2 levels "Female","Male": 2 1 1 2 2 2 2 2 2 2 ...
  ..$ Date.Birth    : Factor w/ 125 levels "01-Jul-26","01-Jul-51",...: 9 2 49 58 35 12 28 10 111 53 ...
  ..$ Weight        : num [1:97] 48 76.1 59.8 56.5 69.9 ...
  ..$ Height        : Factor w/ 40 levels "147","150","151",...: 19 4 11 22 24 24 26 36 31 28 ...
  ..$ BMI           : Factor w/ 90 levels "16","16.6","17.6",...: 2 80 26 6 28 87 59 38 67 75 ...
  ..$ Sample.date   : Factor w/ 109 levels "01-Apr-05","01-Jul-04",...: 66 66 10 49 62 4 17 64 72 79 ...
  ..$ Seizure.type  : Factor w/ 3 levels "IGE","LRE","UNC": 2 2 2 2 1 2 1 1 2 2 ...
  ..$ Simple.Sz.type: Factor w/ 2 levels "IGE","LRE": 2 2 2 2 1 2 1 1 2 2 ...
  ..$ Out.6.m       : int [1:97] 1 2 1 2 1 1 1 1 1 1 2 ...
  ..$ Out.6.m.s     : int [1:97] 1 2 1 2 1 1 1 1 1 1 2 ...
 $ spectra:'data.frame':
  97 obs. of 244 variables:
  ..$ 9.98: num [1:97] 17.7 19.4 18.2 18.9 18.8 ...
  ..$ 9.94: num [1:97] 17.7 19.4 18.2 19 18.8 ...
  ..$ 9.9 : num [1:97] 17.8 19.5 18.2 19 18.9 ...
  ..$ 9.86: num [1:97] 17.9 19.5 18.3 19.1 18.9 ...
  ..$ 9.82: num [1:97] 18 19.6 18.4 19.1 19 ...
  ..$ 9.78: num [1:97] 18.1 19.7 18.5 19.2 19.1 ...
  ..$ 9.74: num [1:97] 18.2 19.8 18.6 19.3 19.2 ...
  ..$ 9.7 : num [1:97] 18.2 19.9 18.6 19.4 19.3 ...
  ..$ 9.66: num [1:97] 18.3 19.9 18.7 19.5 19.3 ...
```

Figure D.2: Example of the patients clinical and spectra information as stored in an R object of class epiData.

standard deviation and median spectra of the data, as well as the covariance matrix. An example of such information stored in an `epiData` list can be seen in Figure D.3. The second object in the list is the mean vector of the spectra,

```

SimGuide: R - Konsole
|R|>str(dSet[[2]])
Named num [1:244] 18.9 19 19 19.1 19.2 ...
- attr(*, "names")= chr [1:244] "9.98" "9.94" "9.9" "9.86" ...
|R|>str(dSet[[3]])
num [1:244, 1:244] 0.398 0.396 0.397 0.397 0.397 ...
- attr(*, "dimnames")=List of 2
..$ : chr [1:244] "9.98" "9.94" "9.9" "9.86" ...
..$ : chr [1:244] "9.98" "9.94" "9.9" "9.86" ...
|R|>str(dSet[[4]])
Named num [1:244] 0.631 0.628 0.629 0.629 0.629 ...
- attr(*, "names")= chr [1:244] "9.98" "9.94" "9.9" "9.86" ...
|R|>str(dSet[[5]])
Named num [1:244] 19 19 19.1 19.1 19.2 ...
- attr(*, "names")= chr [1:244] "9.98" "9.94" "9.9" "9.86" ...

```

**Figure D.3:** Example of the mean, stdev, median and covariance matrix information as stored in an R object of class `epiData`.

the third the covariance matrix, the fourth the standard deviation vector and the fifth the median vector. Pre-treatment of the data is also handled here in two stages:

- **Row scaling to a constant total**, by dividing each sample (row) in the data frame (matrix) by the sum of the values of the variables in each column.
- **Element transformation of the data**, by applying either the *log* or the *square root* transformation to the elements of the data matrix.

Extraction of the unclassified (with respect to their response to AEDs) patients also takes place, before calculating the mean, standard deviation and median vectors. The covariance matrix of the data is converted to positive definite matrix before being stored to the `epiData` matrix.

Function `createDataClass()` was created to read the data from two `.csv` files, and to create from these data an object of class `epiData`. There are ten arguments which take care of the above mentioned operations to the data.

The first two arguments, `sFile` and `cFile`, are the names of the `.csv` files containing the spectral and the clinical information of the patients respectively, if these files are in the current working directory of R, or the full path of the directories in the media the files are stored.

Arguments `dppm` and `uppm` are integer values, representing the range of variables to retain in the data frame. The former argument has three options for its value, of which only the third can be used for data of any number of variables, whereas the other two are suitable only for the data with the 338 variables with 0.04 *ppm* bin width.

- Value 0, means that the first 25 variables, from 10.02 - 10.98 *ppm* chemical shifts will be removed, as well as the water variables (4.70 - 4.90 *ppm*).
- Value -1, means that the first 125 variables, from 6.02 - 10.98 *ppm* chemical shifts will be removed, as well as the water variables (4.70 - 4.90 *ppm*).
- Any other value of `ddpm`, removes the first `dppm` variables, from 10.98 towards 0.02 *ppm* chemical shifts, but does not remove any water variables, as this is needed in some of the analyses.

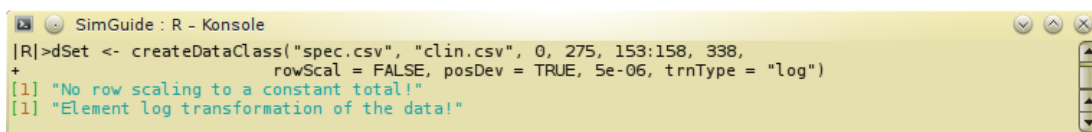
The latter argument is independent of the number of variables in the data set and is the variable (written as column number in the data matrix) above which all variables will be excluded. In the 338 variables data, setting `uppm` to 275, will result in excluding the variables with chemical shifts below 0.02 *ppm*.

Argument `water` is a vector with the range of variables containing water, written as column numbers in the data matrix. In the 338 variables data, the water variables are in columns 153 : 158. The last variable in the data set is stated, as the last column in the data matrix, by argument `fVar`. This is 338 for the data with 338 variables.

With regards to scaling or transforming the data, row-scaling to constant total is applied to the data by setting the `rowScal` argument to `TRUE`. The obtained covariance matrix of the data is converted to positive definite, using the R function `make.positive.definite()` of package `*corpcor*`, by setting argument `posDev` to `TRUE` and the tolerance level argument `tol` to a value such as  $5e - 06$ , to ensure all singular values of the covariance matrix will be positive. The `tol` value is to be selected after examining the covariance matrix and its singular values, and before creating a data frame from the data for the final time. Transformation of the elements in the data matrix is taken care by the `trnType` argument, which can be set to either `log` or `sqrt` for log or square root transformation of the data respectively. If `TrnType` is missing, then no element transformation of the data takes place, during the creation of the data frame. Appropriate text messages are sent to the standard output, to state what type of scaling or transformation has been chosen, each time `createDataClass()` is used to create an object of class `epiData`.

An example of creating an object of class `epiData` can be seen in Figure [D.4](#). In this example, the spectral and the clinical information are contained in the `.csv` files `spec.csv` and `clin.csv` respectively. Both files are in the current working directory of R, therefore there is no need to write down the whole absolute path of the directory in which the files are, in the storage media in use. The





```
SimGuide : R - Konsole
|R|>dSet <- createDataClass("spec.csv", "clin.csv", 0, 275, 153:158, 338,
+                          rowScal = FALSE, posDev = TRUE, 5e-06, trnType = "log")
[1] "No row scaling to a constant total!"
[1] "Element log transformation of the data!"
```

**Figure D.4:** An example of the use of R function `createDataClass()` to obtain an R object of class `epiData`.

patient information is for the 0.04 bin width data with 338 variables. Setting `ddpm` and `uppm` to 0 and 275 respectively, means that the selected data, contains the variables with chemical shifts in the range 0.02 - 9.98 *ppm*, with the water variables (set as columns 153:158) in the range 4.70 - 4.90 *ppm* being removed. The final data frame contains 244 variables in the range mentioned previously. The final variable is in column 338. No row-scaling has been selected, but the covariance matrix is converted to positive definite with tolerance 5e-06. Finally, the data is log transformed, as the message on the standard output states.

## D.2 Generation of Reference and Test Sets

Function `generateSet()` is used to generate a reference and a test set, as required by the simulation experiments. A test set is usually generated after a reference test, as these will constitute the pair of generated tests for the comparisons in a simulation experiment. There are nine arguments which can be set in `generateSet()`. The first argument, `dSet`, is an object of class `epiData`, which is used to generate the two data sets. Arguments `water` and `dppm` are practically the same that were used, as described in the previous section, in `createDataClass()` to create the `dSet` object. These two arguments are used to ensure that `epiData` objects with or without the water variables can be used in `generateSet()`.

Argument `offSet` is a logarithmic value, as the spectral information in `dSet` have been log-transformed and are in logarithmic scale. Its value is 0 if a reference set is required, otherwise it is the value to be added to the mean of a selected for mean-shifting variable, when generating a test set. The number of variables to mean-shift is set by argument `nVars`. This number can be from 1 (case MS1) up to the total number of variables in `dSet`. When generating a reference set, its value should be set to 0. Argument `nRows` is the number of required samples in the reference or test set to be generated. The number of samples in the two sets can be equal or unequal.

Three methods can be used to select the variables for mean-shifting. The required method is controlled by the two arguments `vMethod` and `vOrder`. More

specifically, the available methods include the *standard deviation*, the *mean* and the *median* of the variables, selected by argument `vMethod`. The order (increasing or decreasing) of the variables values with regards to the selected method, is selected by argument `vOrder`. For example, if the required method is to select the `nVars` variables with the maximum standard deviation, `vMethod` must be set to "stdev" and `vOrder` to TRUE (means decreasing is TRUE). In addition, when generating a test set, argument `rSet` is used, corresponding to a character string containing the name of the already generated reference set. If this argument is missing (not used), then a reference set is generated.

An example of the use of `generateSet()`, and the generated sets can be seen in Figure D.5 In this example, the `epiData` object is the one created as seen in

```

SimGuide - R - Konsole
|R|>refSet <- generateSet(dSet, 153:158, 0, 0, 0, 500, "mean", TRUE)
|R|>str(refSet)
List of 3
 $ Values: num [1:500, 1:244] 19 18.7 18.8 19.2 18.3 ...
 .. attr(*, "dimnames")=List of 2
 .. ..$ : chr [1:500] "1" "2" "3" "4" ...
 .. ..$ : chr [1:244] "9.98" "9.94" "9.9" "9.86" ...
 $ Mean : Named num [1:244] 18.9 18.9 19 19.1 19.1 ...
 .. attr(*, "names")= chr [1:244] "9.98" "9.94" "9.9" "9.86" ...
 $ StDev : Named num [1:244] 0.613 0.61 0.612 0.612 0.611 ...
 .. attr(*, "names")= chr [1:244] "9.98" "9.94" "9.9" "9.86" ...
|R|>testSet <- generateSet(dSet, 153:158, 0, 0.4, 120, 500, "mean", TRUE, "refSet")
|R|>str(testSet)
List of 3
 $ Values: num [1:500, 1:244] 19.9 19 19.4 18.2 18.1 ...
 .. attr(*, "dimnames")=List of 2
 .. ..$ : chr [1:500] "501" "502" "503" "504" ...
 .. ..$ : chr [1:244] "9.98" "9.94" "9.9" "9.86" ...
 $ Mean : Named num [1:244] 18.9 18.9 19 19 19.1 ...
 .. attr(*, "names")= chr [1:244] "9.98" "9.94" "9.9" "9.86" ...
 $ StDev : Named num [1:244] 0.659 0.656 0.657 0.657 0.657 ...
 .. attr(*, "names")= chr [1:244] "9.98" "9.94" "9.9" "9.86" ...

```

**Figure D.5:** An example of the use of R function `generateSet()` to create a reference and a test set.

Figure D.4, with the `water` and `dppm` arguments set to 153:158 and 0 respectively. The offset in the generation of the test set is set to 0.4, the number of variables to mean-shift in the test set to 120, the sample size of the two sets to 500, the method of selecting the variables to mean-shift to "mean" in decreasing order (MAXMEAN), and the name of the reference set in the generation of the test set, to "refSet". Function `generateSet()` calculates also the mean and the standard deviation vectors of the two sets and adds the information to the two objects generated for the two data sets (in this case `refSet` and `testSet`).

## D.3 Simulation Analyses

A simulation experiment, as described in Chapter 8, consists of inputting a data set with the original epilepsy data to R, generate two data sets based on the

epilepsy data and perform a series of statistical analyses on the generated data sets, to assess whether the data sets and the selected parameters for the simulation experiments are satisfactory to the requirements of the experiments or not. The most important part of the statistical analyses is taken care by function `simulateData()`. More specifically, there are two main aims when using this function, to either illustrate graphically the potential ability of PCA to discriminate between the two data sets, or to return the information about the misclassification error of LDA and average separation of the two sets, with respect to the selected parameters of the simulation experiment. The first three arguments of `simulateData()`, `dSet`, `rSet` and `tSet` are an object of class `epiData`, a generated by `dSet` reference data set and a similarly generated test set, respectively. Argument `ofValue` is the offset used in the generation of `tSet`. The number of intervals in which to break the range of chemical shifts in any spectrum plot created by `simulateData()`, is set by argument `rppm`. If the spectral data in `dSet` and consequently in `rSet` and `tSet` have been row-scaled to constant total, it is necessary before continuing to any analysis, to pre-treat the data in the generated sets, by re-exponentiating, row-scaling to constant total and log-transforming the data in both sets. Argument `rowScal` ensures that these operations will be done, if set to `TRUE`. If `dSet` is unscaled, then `rowScal` must be set to `FALSE`.

The selection of which of the two main operations of `simulateData()` will be executed is done by setting the argument `simTest` to either `"pca"` for plotting the PCA scores for the two data sets, or `"stat"` to obtain results for the two statistics. An example of the output obtained by `"pca"` can be seen in Figures D.6 and D.7. In this example, the data sets `dSet`, `refSet` and `testSet`, which were produced

```

SimGuide : R - Konsole
|R|>simulateData(dSet, refSet, testSet, 0.4, 10, FALSE, "pca", FALSE, FALSE)
[[1]]
Call:
lda(pcaData$x[, 1:2], grouping = isArtificial, method = "moment")

Prior probabilities of groups:
  0  1
0.5 0.5

Group means:
      PC1      PC2
0  1.434087  1.510445
1 -1.434087 -1.510445

Coefficients of linear discriminants:
      LD1
PC1 -0.04457552
PC2 -1.72620535

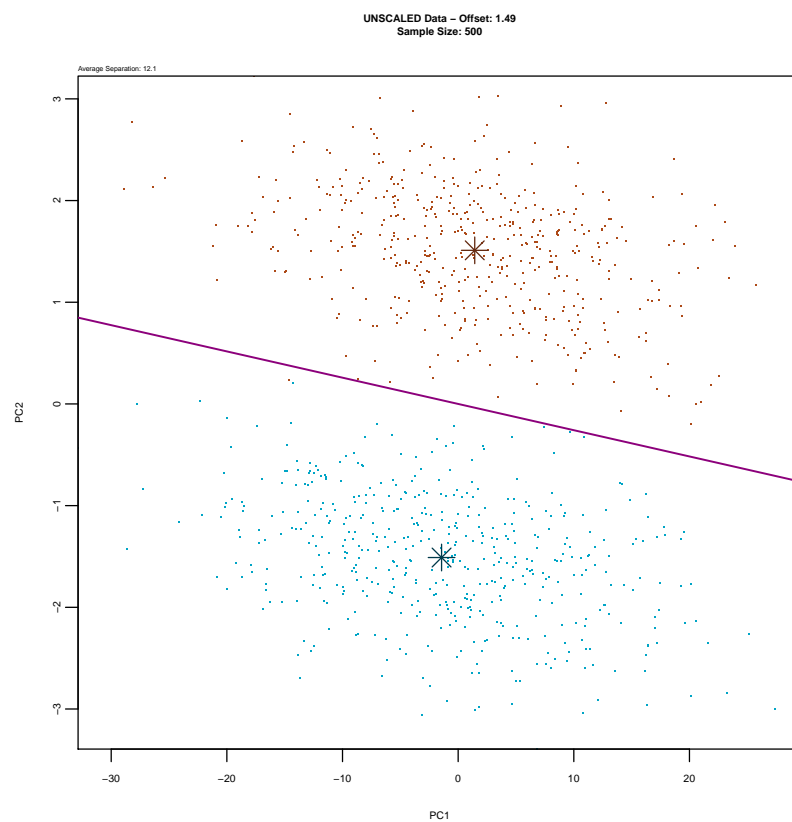
[[2]]
[1] 12.10061

[[3]]
[1] "98.06" "95.34" "2.67" "2.59" "244"

```

**Figure D.6:** Results of a simulation analysis with the use of the R function `simulateData()` for option `"pca"`.

in the examples of the previous sections, have been used. The offset is 0.4, as in the generation of `testSet` previously, `rppm` has been set to 10, to plot the variables in the range 0-10 *ppm*, the data is unscaled, and the last two arguments have been set to their default values `FALSE`. The information returned by "pca" on the standard output, includes in this order, the prior probabilities of the two sets, their group means coordinates and the coefficients of the linear discriminant, as calculated by LDA. In addition, PCA returns the average separation value, the proportion of variance explained by the first PC, the variance of the first PC, the proportion of variance explained by the second PC, the variance of the second PC and the number of PCs that have been retained. In the case of `simTest` being set



**Figure D.7:** PCA scores plot with the superimposed LDA boundary for the analysis in Figure D.6.

to "stat", an object, containing the information about the two statistics and the PCA, is returned. Figure D.8 illustrates such an object, for the same example as in Figure D.6, but with `simTest` set to "stat". In this case, information about the number of classes and the class to which each sample in the two data sets belongs, as well as the values of the two statistics, together with the results of the PCA (as returned also in "pca"), is contained in the produced object.

```

SimGuide: R - Konsole
|R|>simData <- simulateData(dSet, refSet, testSet, 0.4, 10, FALSE, "stat", FALSE, FALSE)
|R|>str(simData)
List of 7
 $ class      : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
 $ posterior  : num [1:1000, 1:2] 1 0.999 1 1 1 ...
 .. attr(*, "dimnames")=List of 2
 .. ..$ : chr [1:1000] "1" "2" "3" "4" ...
 .. ..$ : chr [1:2] "0" "1"
 $ x         : num [1:1000, 1] -2.91 -1.33 -2.95 -3.09 -2.11 ...
 .. attr(*, "dimnames")=List of 2
 .. ..$ : chr [1:1000] "1" "2" "3" "4" ...
 .. ..$ : chr "LD1"
 $ classes   :List of 2
 ..$ ref    : num 0
 ..$ test   : num 1
 $ aversep  : num 12.1
 $ error    : num 0.1
 $ pc       : chr [1:5] "98.06" "95.34" "2.67" "2.59" ...

```

**Figure D.8:** Results of a simulation analysis with the use of the R function `simulateData()` for option "stat".

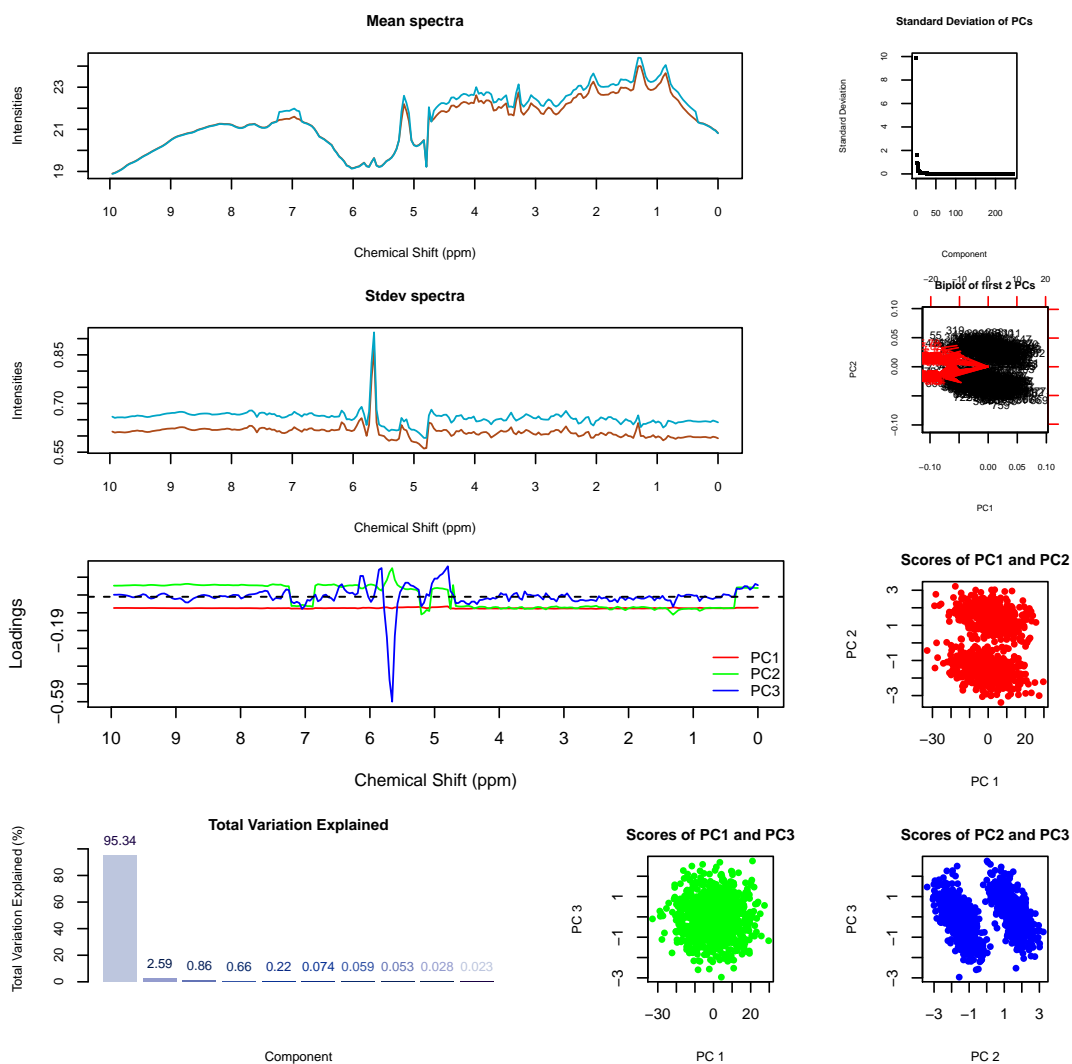
Additional information about the procedure of analysing the data in a simulation experiment, can be seen by setting argument `runTool` to `TRUE`. This will set a call to R function `plotData()`, which produces a detailed graphical output of all results of PCA, as well as of the means of the two data sets. The number of PCs, that `plotData()` plots information about has been set to 3 (the first three PCs), due to space limitations and the structure of the graphical output. An example of the output produced by `plotData()`, for the same arguments as in the "pca" example previously, can be seen in Figure D.9. The last argument in `simulateData()`, if set to `TRUE`, then the PCA information is stored to a `pcaData` object in R's working environment, for further use in other analyses required by the simulation experiments.

## D.4 Execution of the Simulation Algorithm

A simulation experiment requires all steps of the algorithm to be executed as described in Subsection 8.3.8. This is the job of function `runSimulation()`, the *main* function of the simulation algorithm. More specifically, during each experiment, this function calls functions `generateSet()` and `simulateData()`, to generate a pair of sets and run the appropriate statistical analyses, respectively. However, before executing `runSimulation()`, it is necessary to execute `createDataClass()`, in order to obtain the required object of class `epiData`.

A large number of arguments need to be set in order to execute simulation experiments. Argument `dSet` is an object of class `epiData`, as described previously. The `water` argument has been described in detail in the previous sections.

To allow the determination of offsets such that misclassification rates in the range of 20% to 1% can be identified, a range of offsets must be defined and the

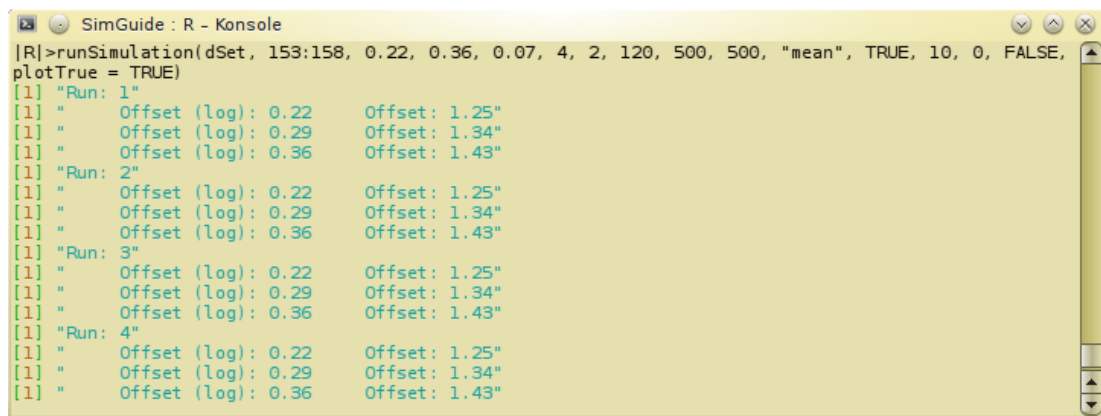


**Figure D.9:** Example of the graphical output of R function `plotData()` for the simulation experiment in Figure D.6.

algorithm executed in a sequence of experiments with fixed step size. For example, if we want in a single run of the algorithm, to perform experiments in the range of offsets 0.4 - 0.7 with step size 0.05, then seven experiments with offsets 0.4, 0.45, 0.5, 0.55, 0.6, 0.65 and 0.7 will take place, and the values of the two statistics, misclassification rate and average separation for each of these experiments will be stored to suitable objects for further analyses and interpretation of the results. Arguments `sValue` and `eValue` represent the first and the last offset, respectively, for which the algorithm will be executed and argument `stepSize`, the increment value which is added in each run of the algorithm to the offset, until offset is at most equal to `eValue`.

The number of runs of the algorithm is set by `nRuns`, and the number of statistics to calculate and store information about, is set by `nPars`. The sample size of the two sets to be generated in an experiment is set by arguments `rSize` and `tSize` for the reference and the test set respectively. The two sets may either have equal sample sizes or unequal. Arguments `nVars`, `vMethod` and `vOrder` are similar to those in `generateSet()`. Similarly, `rppm` and `dppm` are the same as in `simulateData()` and `createDataClass()`. Arguments `rowScal`, `runTool` and `storePCA` are the same as in `simulateData()`. The decision on whether a PC scores plot will be created or the statistics will be calculated is handled by `plotTrue`. If it is set to `TRUE`, then a PC scores plot is produced, otherwise the statistics are returned. This is used, when a call to `simulateData()` is required.

An example of the output returned by `runSimulation()` with `plotTrue` set to `TRUE`, can be seen in Figure D.10. Arguments `dSet`, `water`, `vMethod`, `vOrder`,



```

[R]>runSimulation(dSet, 153:158, 0.22, 0.36, 0.07, 4, 2, 120, 500, 500, "mean", TRUE, 10, 0, FALSE,
plotTrue = TRUE)
[1] "Run: 1"
[1] "   Offset (log): 0.22   Offset: 1.25"
[1] "   Offset (log): 0.29   Offset: 1.34"
[1] "   Offset (log): 0.36   Offset: 1.43"
[1] "Run: 2"
[1] "   Offset (log): 0.22   Offset: 1.25"
[1] "   Offset (log): 0.29   Offset: 1.34"
[1] "   Offset (log): 0.36   Offset: 1.43"
[1] "Run: 3"
[1] "   Offset (log): 0.22   Offset: 1.25"
[1] "   Offset (log): 0.29   Offset: 1.34"
[1] "   Offset (log): 0.36   Offset: 1.43"
[1] "Run: 4"
[1] "   Offset (log): 0.22   Offset: 1.25"
[1] "   Offset (log): 0.29   Offset: 1.34"
[1] "   Offset (log): 0.36   Offset: 1.43"

```

**Figure D.10:** Example of the output of R function `runSimulation()` with `plotTrue` set to `TRUE` and `multiple` to `FALSE`.

`rppm`, `dppm` and `rowScal`, are the same as in the previous examples, being set to `dSet`, `153:158`, `"mean"`, `TRUE`, `10`, `0` and `FALSE` respectively. Starting offset is 0.22 and last offset 0.36 in the logarithmic scale, while a small number of experiments for illustrative purposes has been chosen, hence the step size is set to 0.07, with three experiments in each run taking place. The chosen offsets in the three experiments correspond to misclassification rates 20%, 10% and 1%. The number of runs is set to 4, also for illustrating purposes. The case of mean-shifting 120 variables is chosen, with both generated data sets having in each experiment the same sample size of 500.

It is important to note that, when `plotTrue` is set to `TRUE`, unless only one experiment has been selected, the graphics device driver that has been created for the first scores plot, will just be refreshed each time a new experiment takes place, with the corresponding scores plot taking the place of the existing scores

plot in the device driver, and no new device driver will be started. Therefore, only the scores of the last experiment will be retained at the end of the runs of the algorithm. This can be avoided by setting the argument `multiple` to `TRUE`, and selecting the number of scores plots in a column and in a row of the multiple plot using the arguments `plotRows` and `plotCols` respectively. The same example to that in Figure D.10 but with `multiple` set to `TRUE`, `plotRows` to 4 and `plotCols` to 3 can be seen in Figure D.11.

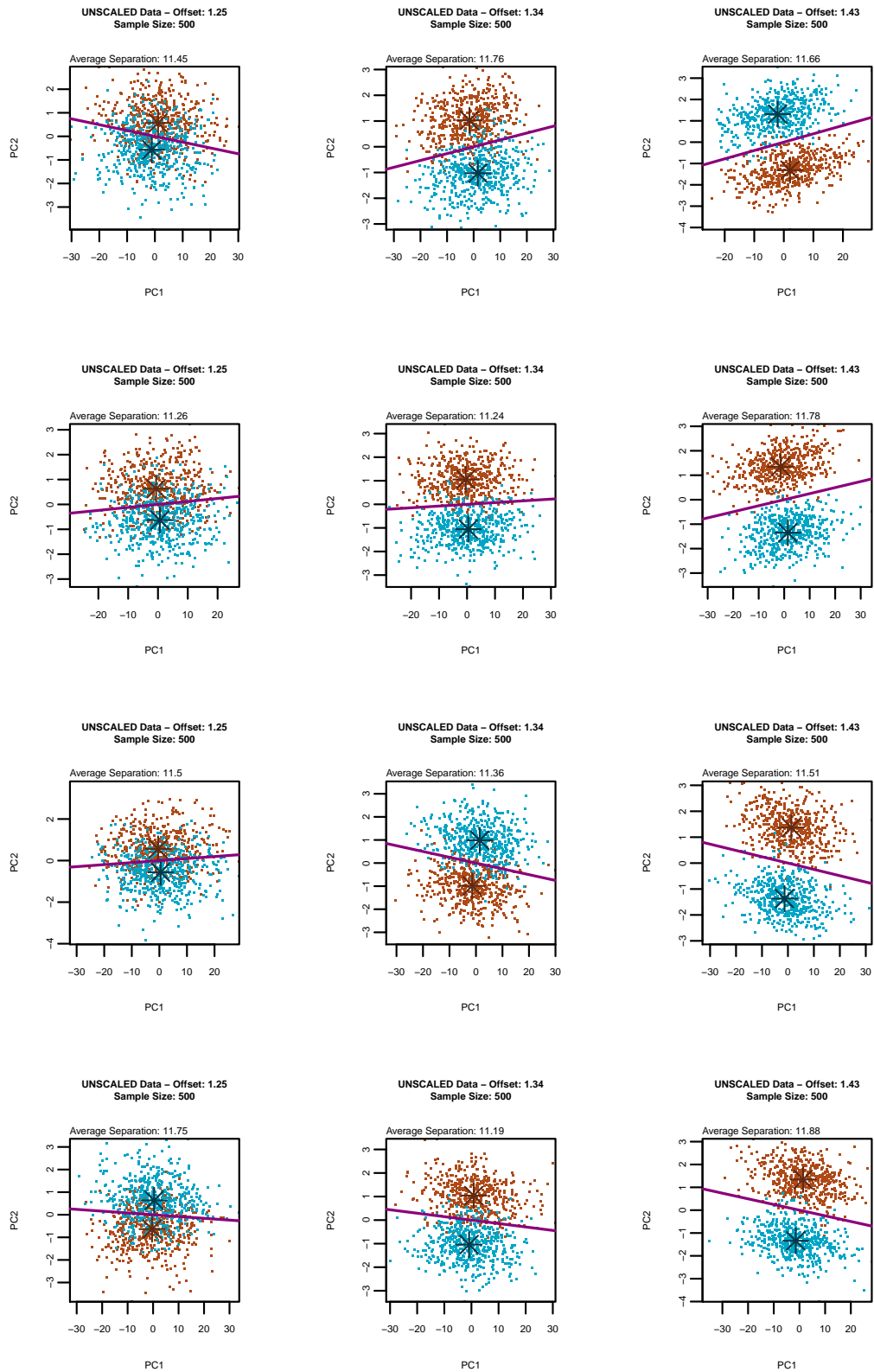
If `plotTrue` is set to `FALSE`, that is, if the results for the two statistics, misclassification rate and average separation are required, then instead of any scores plots, information about each single experiment in every single run of the simulation algorithm is send on the standard output and stored in pre-defined R objects. More specifically, in each experiment, apart from the offset, both in logarithmic and decimal scale (additive value and multiplicative factor respectively), the variances of the first two PCs (VPC1 and VPC2), the proportion of the variance explained by the first two PCs (PC1 and PC2) and the number of PCs retained, are send to the standard output. In addition, an R list is created, containing the average values of the two statistics in each offset for all runs, the analytical tables of the two statistics for each and every run and offset, and a latex table generated by R function `xtable`, containing the average values of the two statistics. Setting `plotTrue` to `FALSE` in the example of Figure D.10 results to the output which can be seen in Figure D.12.

Although the main results of the simulation experiments are given by executing `runSimulation()`, there is other important information to be gathered from each experiment, such as the effect of the mean-shifting to the mean spectra of the two sets, the scores plots for the offsets corresponding to misclassification rates 20%, 10% and 1%, in each experiment case, and the relationship between the two statistics and the offsets in a pre-selected number of runs of the simulation algorithm. The R functions developed with the purpose of gathering and presenting in a graphically way, this information, are described in the following sections.

## D.5 Illustrate the Effect of Mean-shifting

Function `plotMeanShifting()` has been developed to allow the graphical comparison between the mean spectra of the two generated data sets in each experiment. The arguments of this function, include the already seen `dSet`, `water`, `offset`, `nVars`, `vMethod`, `vOrder` and `rppm`. Argument `smpSize` is the sample size





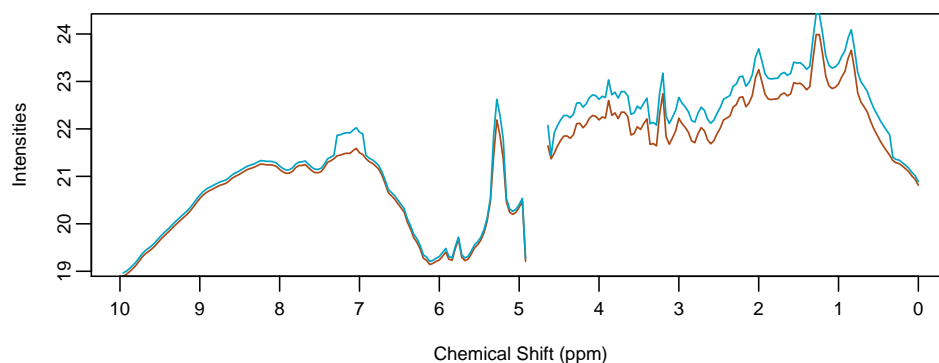
**Figure D.11:** Example of the output of R function `runSimulation()` with `plotTrue` set to `TRUE` and `multiple` to `TRUE`.



of the two generated data sets and it is assumed to be equal in both data sets. It should be noted that `dSet` must not contain any water variables, when using this function, as this function introduces water variables in the mean spectra of the two data sets, to allow for correct plotting of the two mean spectra. This function is applicable to data sets with 244 variables, as in the cases in Chapter 8. Figure D.13 shows the effect of the mean-shifting in one of the cases seen in the previous examples, with the arguments set as previously, apart from the `offset` which is set to 0.36. In addition, the `water` in this case is set to 128:133 due to the water

```
SimGuide : R - Konsole
|R|>plotMeanShifting(dSet, 128:133, 500, 0.36, 120, "mean", TRUE, 10)
```

(a) Command for `plotMeanShifting()`



(b) Mean-shifting effect

**Figure D.13:** Example of the output of R function `plotMeanShifting()`.

adjustments that occur when using this function. The blue and brown colours correspond to the mean spectra of the test and reference data set respectively.

## D.6 Plot PCs Scores and LDA Boundary

In each simulation experiment, a set of three PCs scores plots is produced. The offsets selected for these plots correspond to misclassification rates of 20%, 10% and 1%. This is used, to illustrate how the samples in the two generated data sets are affected by the mean-shifting operation, and to show graphically the actual distance (discriminating ability of PCA) between the two data sets. Arguments `dSet`, `water`, `nVars`, `sSize`, `vMethod`, `vOrder`, `rppm`, `dppm` and `rowScal` have been discussed in the previous sections. The sample size must be the same for both data sets, and it is independent of whether `dSet` contains water variables or

not. In addition, the number of variables in `dSet` is not restricted to 244, as in `plotMeanShifting()`.

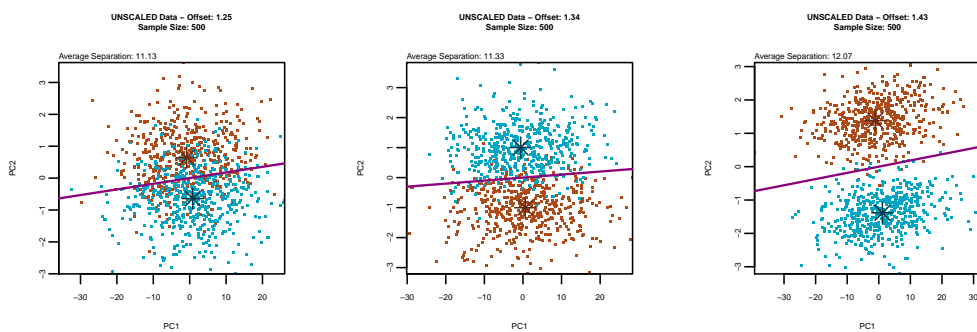
An example of the output of function `plotBoundaries()` can be seen in Figure D.14, for `v1`, `v2` and `v3` set to 0.22, 0.29 and 0.36 respectively. The rest of the arguments are set to similar values to the previous examples. The blue and brown

```

SimGuide : R - Konsole
|R|>plotBoundaries(dSet, 153:158, 0.22, 0.29, 0.36, 120, 500, "mean", TRUE, 10, 0)
[1] "Run: 1"
[1] "   Offset (log): 0.22   Offset: 1.25"
[1] "Run: 1"
[1] "   Offset (log): 0.29   Offset: 1.34"
[1] "Run: 1"
[1] "   Offset (log): 0.36   Offset: 1.43"

```

(a) Command for `plotBoundaries()`



(b) PCs Scores plots

Figure D.14: Example of the output of R function `plotBoundaries()`.

colours correspond to the samples of the test and reference data set respectively. The line is the LDA boundary and the stars are the means of the two data sets.

## D.7 Plot Statistics vs Offsets

Function `plotSimStats()` has been developed to allow the graphical depiction of the relationship between the average values of the two statistics in a pre-selected number of runs and the offsets required to obtain these statistics. In order to use `plotSimStats()` one needs to first obtain an R object of class `statData`, created by `runSimulation()`. Figure D.15 illustrates how this can be done. The information stored in the object `parData`, as well as its structure, can be

```

SimGuide : R - Konsole
|R|>parData <- runSimulation(dSet, 153:158, 0.22, 0.36, 0.07, 4, 2, 120, 500, 500, "mean", TRUE, 10,
0, FALSE, plotTrue = FALSE)

```

Figure D.15: Example of obtaining an object of class `statData`.

seen in Figure D.16. Object `parData` can then be used as the first argument in

```

[R]>parData
[[1]]
      1.25      1.34      1.43
Error Rate (%) 20.80000 8.30000 1.77500
Average Separation 11.15483 11.27477 11.90618
attr(,"class")
[1] "statData"

[[2]]
[[2]]$misrate
      [,1] [,2] [,3]
[1.] 22.3  8.9  1.8
[2.] 20.5  7.3  1.7
[3.] 20.9  7.5  1.8
[4.] 19.5  9.5  1.8

[[2]]$aversep
      [,1] [,2] [,3]
[1.] 11.2470 11.1147 12.2339
[2.] 10.9391 11.4958 11.9632
[3.] 11.4502 11.1736 11.8090
[4.] 10.9830 11.3150 11.6186

[[3]]
% latex table generated in R 2.15.1 by xtable 1.5-6 package
% Wed Feb 13 10:27:57 2013
\begin{table}[ht]
\begin{center}
\begin{tabular}{lccc}
\hline
& 1.25 & 1.34 & 1.43 \\
\hline
Error Rate (\%) & 20.8000 & 8.3000 & 1.7750 \\
Average Separation & 11.1548 & 11.2748 & 11.9062 \\
\hline
\end{tabular}
\end{center}
\end{table}

[R]>str(parData)
List of 3
 $ : statData [1:2, 1:3] 20.8 11.15 8.3 11.27 1.77 ...
 .. attr(*, "dimnames")=List of 2
 .. ..$ : chr [1:2] "Error Rate (%)" "Average Separation"
 .. ..$ : chr [1:3] "1.25" "1.34" "1.43"
 $ :List of 2
 ..$ misrate: num [1:4, 1:3] 22.3 20.5 20.9 19.5 8.9 7.3 7.5 9.5 1.8 1.7 ...
 ..$ aversep: num [1:4, 1:3] 11.2 10.9 11.5 11 11.1 ...
 $ :Classes 'xtable' and 'data.frame': 2 obs. of 3 variables:
 ..$ 1.25: num [1:2] 20.8 11.2
 ..$ 1.34: num [1:2] 8.3 11.3
 ..$ 1.43: num [1:2] 1.77 11.91
 .. attr(*, "align")= chr [1:4] "l" "c" "c" "c"
 .. attr(*, "digits")= num [1:4] 4 4 4 4
 .. attr(*, "display")= chr [1:4] "s" "f" "f" "f"

```

Figure D.16: Example of an object of class `statData`.

`plotSimStats()`. The values of the arguments `sValue`, `eValue` and `stepSize`, are the same that were used to run the experiments in `runSimulation()`, whose results are stored in `parData`. It is also necessary, to load package `*Hmisc*` before running `plotSimStats()`, to allow the use of function `errbar()`, which plots the error bars in the statistics plots. Function `plotSimStats()` calls another function, `createStatsPlots()`, which was developed to create the three plots, with regards to the information stored in `parData`.

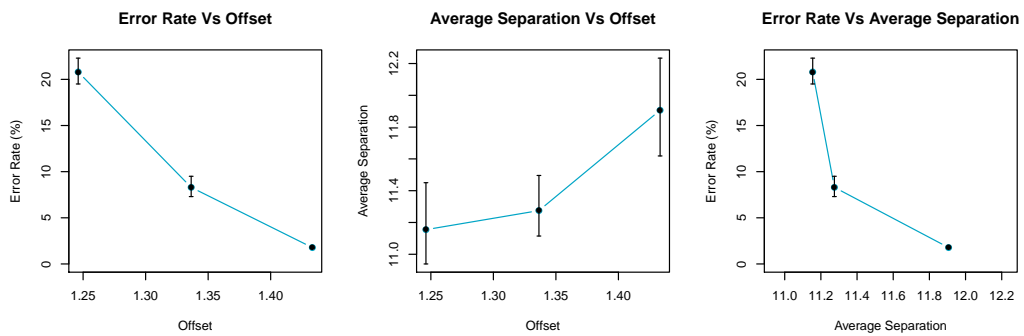
An example of the output of `plotSimStats()`, using the object `parData`, `sValue` set to 0.22, `eValue` to 0.36 and `stepSize` to 0.07, can be seen in Figure D.17.

```

SimGuide : R - Konsole
|R|>plotSimStats(parData, 0.22, 0.36, 0.07)

```

(a) Command for plotSimStats()



(b) Stats vs Offset plots

Figure D.17: Example of the output of R function plotSimStats().

## D.8 Plot Additional Information

As was described in Section 9.2, and shown in Figure D.9, function plotData() provides additional information for the simulation experiments. This is an internal function, in the sense that it is usually executed through runSimulation() and simulateData(). That is mainly because two of the required arguments in plotData(), bdSet and pcData, are R objects obtained during the execution of simulateData(). It is therefore necessary to execute simulateData() before running plotdata(). Object bdSet is automatically created and stored in R's working environment by simulateData(), but to obtain the object pcData, argument storePCA must be set to TRUE in simulateData().

An example of the structure of these two objects, for the dSet used in the previous examples and the experiments in Figure D.12, can be seen in Figures D.18 and D.19 respectively. The object bdSet contains the row-binded set of the

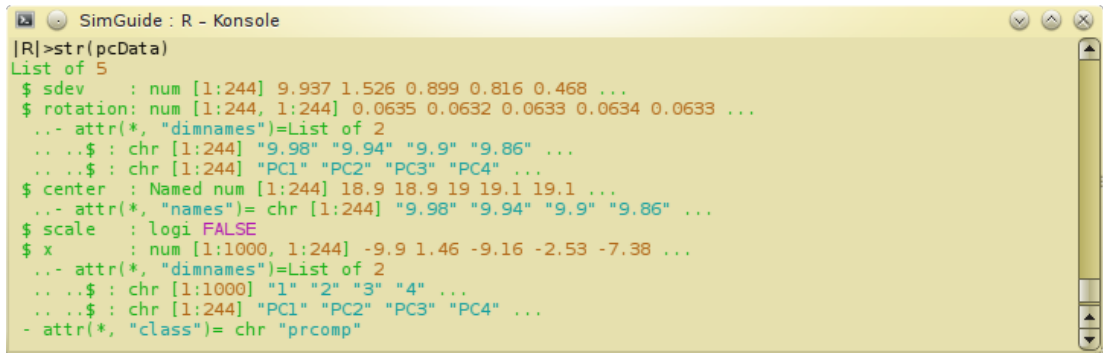
```

SimGuide : R - Konsole
|R|>str(bdSet)
rBindData [1:1000, 1:244] 19.1 19.7 19.9 18.2 19.1 ...
- attr(*, "dimnames")=List of 2
..$ : chr [1:1000] "1" "2" "3" "4" ...
..$ : chr [1:244] "9.98" "9.94" "9.9" "9.86" ...

```

Figure D.18: Example of the structure of a bdSet object.

two generated sets and pcData the results of the PCA for the current experiment. The R function prcomp() is used to perform the PCA, hence the object pcData is of class prcomp().



```
[R]>str(pcData)
List of 5
 $ sdev      : num [1:244] 9.937 1.526 0.899 0.816 0.468 ...
 $ rotation: num [1:244, 1:244] 0.0635 0.0632 0.0633 0.0634 0.0633 ...
 .. attr(*, "dimnames")=List of 2
 .. ..$ : chr [1:244] "9.98" "9.94" "9.9" "9.86" ...
 .. ..$ : chr [1:244] "PC1" "PC2" "PC3" "PC4" ...
 $ center   : Named num [1:244] 18.9 18.9 19 19.1 19.1 ...
 .. attr(*, "names")= chr [1:244] "9.98" "9.94" "9.9" "9.86" ...
 $ scale    : logi FALSE
 $ x        : num [1:1000, 1:244] -9.9 1.46 -9.16 -2.53 -7.38 ...
 .. attr(*, "dimnames")=List of 2
 .. ..$ : chr [1:1000] "1" "2" "3" "4" ...
 .. ..$ : chr [1:244] "PC1" "PC2" "PC3" "PC4" ...
 - attr(*, "class")= chr "prcomp"
```

**Figure D.19:** Example of the structure of a pcData object.

A number of plotting functions are called by `plotData()` to plot the various objects and results of the simulation experiments. These internal functions are `plotMeanSpectrum()`, which plots the mean spectra of the data sets, `plotLoads()`, which plots the PCs loadings vs variables, `plotVarExpl()`, which plots the variance explained by the first ten PCs and function `plotScores()`, which plots the scores for the first three PCs in all combinations of pairs. Object `bdSet` is also used as an argument of `plotLoads()`, whereas object `pcData` is used as an argument of `plotVarExpl()`.

# Appendix E

---

## Components of Mass Spectrometers

### Contents

1. List of *ionisation* methods
2. List of *mass analysers*
3. List of MS *detectors*



**Table E.1:** Comparison of *ionisation sources*

Source	Typical Mass Range (Da)	Sensitivity	Type of ionisation
Electron	500	picomole	Hard
Chemical	500	picomole	Hard
FAB	7,000	nanomole	Semi-hard
ESI	70,000	high femtomole - low picomole	Soft
MALDI	300,000	low to high femtomole	Soft

**Table E.2:** Comparison of *mass analysers*

Analyser	Accuracy (p.p.m.)	Resolution	$\frac{m}{z}$	Range
Quadrupole	100	4,000		4,000
Time-of-flight	200	8,000	>	300,000
Ion Trap	100	4,000		4,000
Magnetic Sector	< 5	30,000		10,000
FTMS	< 5	100,000		10,000

**Table E.3:** Comparison of the most commonly used *detectors*

Detector	Advantages	Disadvantages
Faraday Cup	Good for checking ion transmission and low sensitivity measurements	Low amplification (approximately 10)
Photomultiplier Conversion Dynode (PCD)	-Robust -Long lifetime (> 5 years) -Sensitive (approximately gains of $10^6$ )	Cannot be exposed to light while in operation
Electron Multiplier (SEM)	-Robust -Fast response -Sensitive (approximately gains of $10^6$ )	Shorter lifetime than PCD (around 3 years)
Charge Detection	Detects ions independent of mass and velocity	Limited compatibility with most existing instruments

# Bibliography

- Adams, M. (2004). *Chemometrics in Analytical Spectroscopy*. RSC Analytical Spectroscopy Monographs. Royal Society of Chemistry, Cambridge, UK, second edition.
- Ahmad, N., Alahakoon, D., and Chau, R. (2010). Cluster identification and separation in the growing self-organizing map: application in protein sequence classification. *Neural Computing and Applications*, 19(4):531–542.
- Antti, H., Holmes, E., and Nicholson, J. (2002). Multivariate solutions to metabolic profiling and functional genomics. Part 1 - introduction, data acquisition and processing. Available at: <http://www.acc.umu.se/~tnkjtg/Chemometrics/Editorial>. Last visited on 2010-04-14.
- Barwick, V., Langley, J., Mallet, T., Stein, B., and Webb, K. (2006). Best practice guide for generating mass spectra. Best practice guide, LGC, Teddington, UK.
- Bauer, H. U., Villmann, T., and Herrmann, M. (1999). Neural maps and topographic vector quantization. *Neural Networks*, 12(4-5):659–676.
- Beckonert, O., Bollard, M., Ebbels, T., Keun, H., Antti, H., Holmes, E., Lindon, J., and Nicholson, J. (2003). NMR-based metabonomic toxicity classification: hierarchical cluster analysis and k-nearest-neighbour approaches. *Analytica Chimica Acta*, 490(1-2):3–15.
- Beebe, K., Pell, R., and Seasholtz, M. (1998). *Chemometrics: A Practical Guide*. Wiley-Interscience Series on Laboratory Automation. John Wiley and Sons, New York, USA.
- Berg, R., Hoefsloot, H., Westerhuis, J., and Smilde, A.K. Werf, M. (2006). Centering, scaling and transformations: improving the biological information content of metabolomics data. *BMC Genomics*, 7(6):142–156.
- Besse, P. (1992). PCA stability and choice of dimensionality. *Statistics & Probability Letters*, 13(5):405–410.

- Besse, P. and Falguerolles, A. (1993). Application of resampling methods to the choice of dimension in principal components analysis. In Hardle, W. and Simar, L., editors, *Computer Intensive Methods in Statistics*, Statistics and Computers, pages 167–177. Physica-Verlag, Heidelberg, Germany.
- Bishop, C. (1997). *Neural Networks for Pattern Recognition*. Oxford University Press, New York, USA.
- Blume, W., Luders, H., Mizrahi, E., Tassinari, C., Boas, W., and Engel, J. (2001). Glossary of descriptive terminology for ictal semiology: Report of the ILAE task force on classification and terminology. *Epilepsia*, 42(9):1212–1218.
- Boccard, J., Veuthey, J., and Rudaz, S. (2010). Knowledge discovery in metabolomics: An overview of MS data handling. *Journal of Separation Science*, 33(3):290–304.
- Bollard, M., Holmes, E., Lindon, J., Mitchell, S., Branstetter, D., Zhang, W., and Nicholson, J. (2001). Investigations into biochemical changes due to diurnal variation and oestrus cycle in female rats using high resolution  $^1\text{H}$  NMR spectroscopy of urine and pattern recognition. *Analytical Biochemistry*, 295(2):194–202.
- Bollard, M., Keun, H., Ebbels, T., Beckonert, O., Antti, H., Lindon, J., and Nicholson, J. (2005a). Comparative metabonomics of differential species toxicity of hydrazine in the rat and mouse. *Toxicological Applied Pharmacology*, 204(2):135–151.
- Bollard, M., Stanley, E., Lindon, J., Nicholson, J., and Holmes, E. (2005b). NMR-based metabonomic approaches for evaluating physiological influences on biofluid composition. *NMR in Biomedicine*, 18(3):143–162.
- Borcard, D., Gillet, F., and Legendre, P. (2011). *Numerical Ecology with R*. Use R! Springer, New York, USA.
- Borman, S., Russell, H., and Siuzdak, G. (2003). A mass spec timeline. *Today's Chemist at Work*, pages 47–49.
- Bouchereau, A., Guenot, P., and Larher, F. (2000). Analyses of amines in plant materials. *Journal of Chromatography B*, 747(1-2):49–67.
- Brereton, R. (2009). *Chemometrics for Pattern Recognition*. John Wiley and Sons, West Sussex, UK.

- Brisdon, A. (2003). *Inorganic Spectroscopic Methods*, volume 62 of *Oxford Chemistry Primers*. Oxford University Press, Oxford, GB.
- Cangelosi, R. and Goriely, A. (2007). Component retention in principal component analysis with application to cDNA microarray data. *Biology Direct*, 2(2):1–21.
- Cattell, R. (1966). The scree test for the number of factors. *Multivariate Behavioral Research*, 1(2):245–276.
- Chan, C., Hsu, A., Tang, S., and Halgamuge, S. (2008). Using growing self-organizing maps to improve the binning process in environmental whole-genome shotgun sequencing. *Journal of Biomedicine and Biotechnology*, 2008. Article ID 513701.
- Chan, E., Koh, P., Mal, M., Cheah, P., Eu, K., Backshall, A., Cavill, R., Nicholson, J., and Keun, H. (2009). Metabolic profiling of human colorectal cancer using HR-MAS NMR and GC-MS. *Journal of Proteome Research*, 8(1):352–362.
- Cominetti, O., Matzavinos, A., Samarasinghe, S., Kulasiri, D., Liu, S., Maini, P., and Erban, R. (2010). DiffFUZZY: A fuzzy clustering algorithm for complex data sets. *International Journal of Computational Intelligence in Bioinformatics and Systems Biology*, 1(4):402–417.
- Cox, T. and Cox, M. (2001). *Multidimensional Scaling*, volume 88 of *Monographs on Statistics and Applied Probability*. Chapman and Hall/CRC, Florida, USA, third edition.
- Craig, A., Holmes, E., Nicholson, J., and Lindon, J. (2006). Scaling and normalization effects in NMR spectroscopic metabonomic data sets. *Analytical Chemistry*, 78(7):2262–2267.
- Croux, C., Filzmoser, P., and Oliveira, M. (2007). Algorithms for projection-pursuit robust principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 87(2):218–225.
- Culf, M., Belacel, N., Culs, A., Chute, I., Ouellette, R., Burton, I., Karakach, T., and Walter, J. (2009). NMR metabolic analysis of samples using fuzzy k-means clustering. *Magnetic Resonance in Chemistry*, 47(Suppl 1):S96–S104.

- Daniel, L. (1992). Bootstrap methods in the principal components case. In *Proc. Annual Meeting of the American Educational Research Association*, San Francisco, USA. AERA.
- Devinsky, O. (1999). Patients with refractory seizures. *The New England Journal of Medicine*, 340(20):1565–1570.
- Diamantaras, K. and Kung, S. (1996). *Principal Component Neural Networks: Theory and Applications*. Adaptive and Learning Systems for Signal Processing, Communications and Control. John Wiley and Sons, New York, USA.
- Dimitriadou, E., Dolnicar, S., and Weingessel, A. (2002). An examination of indexes for determining the number of clusters in binary data sets. *Psychometrika*, 67(3):137–160.
- Dittenbach, M., Rauber, A., and Merkl, D. (2002). Uncovering hierarchical structure in data using the growing hierarchical self-organizing map. *Neurocomputing*, 48(1-4):199–216.
- Dixon, S., Xu, Y., Brereton, R., Soini, H., Novotny, M., Oberzaucher, E., Grammer, K., and Penn, D. (2007). Pattern recognition of gas chromatography mass spectrometry of human volatiles in sweat to distinguish the sex of subjects and determine potential discriminatory marker peaks. *Chemometrics and Intelligent Laboratory Systems*, 87(2):161–172.
- Duckett, S. and Gilbert, B. (2002). *Foundations of Spectroscopy*, volume 78 of *Oxford Chemistry Primers*. Oxford University Press, Oxford, GB.
- Ebbels, T. (2007). Non-linear methods for the analysis of metabolic profiles. In Lindon, J., Nicholson, J., and Holmes, E., editors, *The Handbook of Metabonomics and Metabolomics*, chapter 7, pages 201–226. Elsevier, Amsterdam, The Netherlands.
- Eilers, P. (2004). Parametric time warping. *Analytical Chemistry*, 76(2):404–411.
- Ekins, S., Nikolsky, Y., and Nikolskaya, T. (2005). Techniques: Application of systems biology to absorption, distribution, metabolism, excretion and toxicity. *Trends in Pharmacological Sciences*, 26(4):202–209.
- Emsley, J. and Feeney, J. (2007). Forty years of progress in nuclear magnetic resonance spectroscopy. *Progress in Nuclear Magnetic Resonance Spectroscopy*, 50(4):179–198.

- Engel, J. (2006a). ILAE classification of epilepsy syndromes. *Epilepsy Research*, 70(Suppl):5–10.
- Engel, J. (2006b). Report of the ILAE classification core group. *Epilepsia*, 47(9):1558–1568.
- Everitt, B. (1993). *Cluster Analysis*. Arnold, London, UK, third edition.
- Everitt, B. and Hothorn, T. (2006). *A Handbook of Statistical Analyses Using R*. Chapman and Hall/CRC, Florida, USA.
- Everitt, B. and Rabe-Hesketh, S. (1997). *The Analysis of Proximity Data*. Series: Kendall's Library of Statistics 4. Arnold, London, UK.
- Ferre, L. (1995). Selection of components in principal component analysis: A comparison of methods. *Computational Statistics & Data Analysis*, 19(6):669–682.
- Fiehn, O. (2001). Combining genomics, metabolome analysis and biochemical modelling to understand metabolic networks. *Comparative and Functional Genomics*, 2(3):155–168.
- Fiehn, O. (2002). Metabolomics - the link between genotypes and phenotypes. *Plant Molecular Biology*, 48(1-2):155–171.
- Fisher, R., Boas, W., Blume, W., Elger, C., Genton, P., Lee, P., and Engel, J. (2005). Epileptic seizures and epilepsy: definitions proposed by the International League Against Epilepsy (ILAE) and the International Bureau for Epilepsy (IBE). *Epilepsia*, 46(4):470–472.
- Gamache, P., Meyer, D., Granger, M., and Acworth, I. (2004). Metabolomic applications of electrochemistry/mass spectrometry. *Journal of American Society for Mass Spectrometry*, 15(12):1717–1726.
- Gavaghan, C., Nicholson, J., Connor, S., Wilson, I., Wright, B., and Holmes, E. (2001). Directly coupled high-performance liquid chromatography and NMR spectroscopic with chemometric studies on metabolic variation in Sprague-Dawley rats. *Analytical Biochemistry*, 291(2):245–252.
- Gavaghan, C., Wilson, I., and Nicholson, J. (2002). Physiological variation in metabolic phenotyping and functional genomic studies: use of orthogonal signal correction and PLS-DA. *FEBS Letters*, 530(1-3):191–196.

- Gemperline, P. (2006). Principal component analysis. In Gemperline, P., editor, *Practical Guide to Chemometrics*, chapter 4, pages 69–104. Taylor & Francis, New York, second edition.
- Glish, G. and Vachet, R. (2003). The basics of mass spectrometry in the twenty-first century. *Nature Reviews: Drug Discovery*, 2(2):140–150.
- Goodacre, R., Broadhurst, D., Smilde, A., Kristal, B., Baker, J., Beger, R., Bessant, C., Connor, S., Capuani, G., Craig, A., Ebbels, T., Kell, D., Manetti, C., Newton, J., Paternostro, G., Somorjai, R., Sjostrom, M., Trygg, J., and Wulfert, F. (2007). Proposed minimum reporting standards for data analysis in metabolomics. *Metabolomics*, 3(3):231–241.
- Goodacre, R., Vaidyanathan, S., Dunn, W., Harrigan, G., and Kell, D. (2004). Metabolomics by numbers: acquiring and understanding global metabolite data. *Trends in Biotechnology*, 22(5):245–252.
- Gordon, A. (1981). *Classification*. Monographs on Applied Probability and Statistics. Chapman and Hall, London, UK.
- Gordon, A. (1996). Hierarchical classification. In Arabie, P., Hubert, L., and De Soete, G., editors, *Clustering and Classification*, pages 65–121. World Scientific Publishing, River Edge, New Jersey.
- Griffin, J. (2004). The potential of metabonomics in drug safety and toxicology. *Drug Discovery Today: Technologies*, 1(3):285–293.
- Griffin, J., Walker, L., Garrod, S., Holmes, E., Shore, R., and Nicholson, J. (2000). NMR spectroscopy based metabonomic studies on the comparative biochemistry of the kidney and urine of the bank vole (*Clethrionomys glareolus*), wood mouse (*Apodemus sylvaticus*), white toothed shrew (*Crocidura suaveolens*) and the laboratory rat. *Comparative Biochemistry and Physiology Part B: Biochemistry and Molecular Biology*, 127(3):357–367.
- Griffin, J., Williams, H.J. Sang, E., Clarke, K., Rae, C., and Nicholson, J. (2001). Metabolic profiling of genetic disorders: a multitissue  $^1\text{H}$  NMR Spectroscopic and pattern recognition study into dystrophic tissue. *Analytical Biochemistry*, 293(1):16–21.
- Groenen, P. and de Velden, M. V. (2004). Multidimensional scaling. Econometric Institute Report EI 2004-15, Econometric Institute, Erasmus University Rotterdam, Rotterdam, The Netherlands.

- Guadagnoli, E. and Velicer, W. (1988). Relation of sample size to the stability of component patterns. *Psychological Bulletin*, 103(2):265–275.
- Harrigan, G., LaPlante, R., Cosma, G., Cockerell, G., Goodacre, R., Maddox, J., Luyendyk, J., Ganey, P., and Roth, R. (2004). Application of high-throughput Fourier-transform infrared spectroscopy in toxicology studies: contribution to a study on the development of an animal model for idiosyncratic toxicity. *Toxicology Letters*, 146(3):197–205.
- Hartigan, J. (1975). *Clustering Algorithms*. John Wiley & Sons, New York, USA.
- Hartigan, J. and Wong, M. (1979). A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108.
- Hebert, P., Masson, M., and Denoeux, T. (2006). Fuzzy multidimensional scaling. *Computational Statistics & Data Analysis*, 51(1):335–359.
- Hitiris, N., Mohanraj, R., Norrie, J., Sills, G., and Brodie, M. (2007). Predictors of pharmaco-resistant epilepsy. *Epilepsy Research*, 75(2-3):192–196.
- Holmes, E., Bonner, F., Sweatman, B., Lindon, J., Beddell, C., Rahr, E., and Nicholson, J. (1992). Nuclear magnetic resonance spectroscopy and pattern recognition analysis of the biochemical processes associated with the progression of and recovery from nephrotoxic lesions in the rat induced by mercury(II) chloride and 2-bromoethanamine. *Molecular Pharmacology*, 42(5):922–930.
- Holmes, E., Nicholls, A., Lindon, J., Connor, S., Connelly, J., Haselden, J., Damment, S., Spraul, M., Neidig, P., and Nicholson, J. (2000). Chemometric models for toxicity classification based on NMR spectra of biofluids. *Chemical Research in Toxicology*, 13(6):471–478.
- Horn, J. (1965). A rationale and test for the number of factors in factor analysis. *Psychometrika*, 30(2):179–185.
- Hsu, A., Tang, S., and Halgamuge, S. (2003). An unsupervised hierarchical dynamic self-organizing approach to cancer class discovery and marker gene identification in microarray data. *Bioinformatics*, 19(16):2131–2140.
- Hubert, M., Rousseeuw, P., and Branden, K. (2005). ROBPCA: A new approach to robust principal component analysis. *Technometrics*, 47(1):64–79.
- Idborg, H., Zamani, L., Edlund, P., Koistinen, I., and Jacobsson, S. (2005). Metabolic fingerprinting of rat urine by LC/MS Part 2. Data pretreatment



- methods for handling of complex data. *Journal of Chromatography B*, 828(1-2):14–20.
- Izenman, A. (2008). *Modern Multivariate Statistical Techniques - Regression, Classification and Manifold Learning*. Springer Texts in Statistics. Springer, New York, USA.
- Jackson, J. (2003). *A User's Guide to Principal Components*. Wiley Series in Probability and Statistics. Wiley-Interscience, Hoboken, New Jersey.
- Jambu, M. (1978). *Classification Automatique Pour L'analyse des Donnees (Tome 1)*. Dunod, Paris, France.
- Jin, H., Shum, W., Leung, K., and Wong, M. (2004). Expanding self-organizing map for data visualization and cluster analysis. *Information Sciences*, 163(1-3):157–173.
- Kalteh, A., Hjorth, P., and Berndtsson, R. (2008). Review of the self-organizing map (SOM) approach in water resources: Analysis, modelling and application. *Environmental Modelling & Software*, 23(7):835–845.
- Kaufman, L. and Rousseeuw, P. (2005). *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley Series in Probability and Mathematical Statistics. Applied probability and statistics. John Wiley and Sons, New York, USA.
- Kealey, D. and Haines, P. (2002). *Analytical Chemistry*. Instant Notes. BIOS Scientific Publishers, Oxford, UK.
- Keun, H. (2006). Metabonomics modeling of drug toxicity. *Pharmacology & Therapeutics*, 109(1-2):92–106.
- Kiviluoto, K. (1996). Topology preservation in self-organizing maps. In *Proceedings of the IEEE International Conference on Neural Networks, Volume 1*, pages 294–299.
- Krzanowski, W. (1987). Selection of variables to preserve multivariate data structure using principal components. *Applied Statistics*, 36(1):22–33.
- Krzanowski, W. and Marriott, F. (1995). *Multivariate Analysis*. Kendall's Library of Statistics 2. Arnold, London, UK.
- Kwan, P. and Brodie, M. (2000a). Early identification of refractory epilepsy. *The New England Journal of Medicine*, 342(5):314–319.

- Kwan, P. and Brodie, M. (2000b). Epilepsy after the first drug fails: substitution or add-on? *Seizure*, 9(7):464–468.
- Lance, G. and Williams, W. (1967). A general theory of classificatory sorting strategies - 1. hierarchical systems. *The Computer Journal*, 9(4):373–380.
- Lauridsen, M., Bliddal, H., Christensen, R., Samsøe, B., Bennett, R., Keun, H., Lindon, J., Nicholson, J., Dorff, M., Jaroszewski, J., Hansen, S., and Cornett, C. (2010). <sup>1</sup>H NMR spectroscopy-based interventional metabolic phenotyping: A cohort study of rheumatoid arthritis patients. *Journal of Proteome Research*, 9(9):4545–4553.
- Legendre, L. and Legendre, P. (1998). *Numerical Ecology*, volume 20 of *Developments in Environmental Modelling*. Elsevier Science B.V., Amsterdam, The Netherlands, second english edition.
- Lei, R., Wu, C., Yang, B., Ma, H., Shi, C., Wang, Q., Wang, Q., Yuan, Y., and Liao, M. (2008). Integrated metabolomic analysis of the nano-sized copper particle-induced hepatotoxicity and nephrotoxicity in rats: A rapid *in vivo* screening method for nanotoxicity. *Toxicology and Applied Pharmacology*, 232(2):292–301.
- Li, X., Lu, X., Tian, J., Gao, P., Kong, H., and Xu, G. (2009). Application of fuzzy c-means clustering in data analysis of metabolomics. *Analytical Chemistry*, 81(11):4468–4475.
- Lin, Y., Si, D., Zhang, Z., and Liu, C. (2009). An integrated metabonomic method for profiling of metabolic changes in carbon tetrachloride induced rat urine. *Toxicology*, 256(3):191–200.
- Lindon, J. (2004). *Metabonomics - techniques and applications*. Business Briefing: Future Drug Discovery. Metabometrix Ltd., London, UK.
- Lindon, J., Holmes, E., Bollard, M., Stanley, E., and Nicholson, J. (2004). Metabonomics technologies and their applications in physiological monitoring, drug safety assessment and disease diagnosis. *Biomarkers*, 9(1):1–31.
- Lindon, J., Holmes, E., and Nicholson, J. (2001). Pattern recognition methods and applications in biomedical magnetic resonance. *Progress in Nuclear Magnetic Resonance Spectroscopy*, 39(1):1–40.

- Lindon, J., Holmes, E., and Nicholson, J. (2006). Metabonomics techniques and applications to pharmaceutical research and development. *Pharmaceutical Research*, 23(6):1075–1088.
- Lindon, J., Nicholson, J., and Everett, J. (1999). NMR spectroscopy of biofluids. *Annual Reports on NMR Spectroscopy*, 38:1–88.
- Lindon, J., Nicholson, J., Holmes, E., Antti, H., Bollard, M., Keun, H., Beckonert, O., Ebbels, T., Reily, M., Robertson, D., Stevens, G., Luke, P., Breau, A., Cantor, G., Bible, R., Niederhauser, U., Senn, H., Schlotterbeck, G., Sidemann, U., Laursen, S., Tymiak, A., Car, B., McKeeman, L., Colet, J., and Thomas, C. (2003). Contemporary issues in toxicology - the role of metabonomics in toxicology and its evaluation by the COMET project. *Toxicology and Applied Pharmacology*, 187(3):137–146.
- Lindon, J., Nicholson, J., Holmes, E., and Everett, J. (2000). Metabonomics: metabolic processes studied by NMR spectroscopy of biofluids. *Concepts in Magnetic Resonance*, 12(5):289–320.
- Lloyd, S. (1982). Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137.
- Loscher, W. (2002). Current status and future directions in the pharmacotherapy of epilepsy. *Trends in Pharmacological Sciences*, 23(3):113–118.
- Loscher, W. and Schmidt, D. (2002). New horizons in the development of antiepileptic drugs. *Epilepsy Research*, 50(1-2):3–16.
- Lowenstein, D. (2008). Pathways to discovery in epilepsy research: Rethinking the quest for cures. *Epilepsia*, 49(1):1–7.
- Lukasova, A. (1979). Hierarchical agglomerative clustering procedure. *Pattern Recognition*, 11(6):365–381.
- MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In Le Cam, L. and Neyman, J., editors, *Mathematical Statistics and Probability, Proceedings of the 5th Berkeley Symposium, June 21 - July 18, 1965*, Volume I: Theory of Statistics, pages 281–297, Statistical Laboratory, University of California, USA. University of California Press, USA.
- Makinen, V., Soininen, P., Forsblom, C., Parkkonen, M., Ingman, P., Kaski, K., Groop, P., and Korpela, M. (2008).  $^1\text{H}$  NMR metabonomics approach to the

- disease continuum of diabetic complications and premature death. *Molecular Systems Biology*, 4(Article number 167). Available at: <http://onlinelibrary.wiley.com/doi/10.1038/msb4100205/pdf>. Last visited on 2014-10-03.
- Mariey, L., Signolle, J., Amiel, C., and Travert, J. (2001). Discrimination, classification, identification of microorganisms using FTIR spectroscopy and chemometrics. *Vibrational Spectroscopy*, 26(2):151–159.
- Massart, D., Vandeginste, B., Deming, S., Michotte, Y., and Haines, P. (1990). *Chemometrics: a Textbook*, volume 2 of *Data Handling in Science and Technology*. Elsevier, Amsterdam, The Netherlands.
- Milligan, G. and Cooper, M. (1985). An examination of procedures for determining the number of clusters in a data set. *Psychometrika*, 50(2):159–179.
- Myatt, G. (2007). *Making Sense of Data. A Practical Guide to Exploratory Data Analysis and Data Mining*. John Wiley & Sons, New Jersey, USA.
- Neme, A. and Miramontes, P. (2005). Statistical properties of lattices affect topographic error in self-organizing maps. In *Artificial Neural Networks: Biological Inspirations - ICANN 2005*, Lecture Notes in Computer Science, pages 427–432. Springer, Heidelberg, Berlin.
- Neto, P., Jackson, D., and Somers, K. (2005). How many principal components? Stopping rules for determining the number of non-trivial axes revisited. *Computational Statistics & Data Analysis*, 49(4):974–997.
- Nicholson, J., Holmes, E., and Lindon, J. (2007). Metabonomics and metabolomics techniques and their applications in mammalian systems. In Lindon, C., Nicholson, J., and Holmes, E., editors, *Handbook of Metabonomics and Metabolomics*, chapter 1, pages 1–33. Elsevier, Amsterdam, The Netherlands.
- Nielsen, J. and Oliver, S. (2005). The next wave in metabolome analysis. *Trends in Biotechnology*, 23(11):544–546.
- Oberemm, A., Onyon, L., and Gundert-Remy, U. (2005). How can toxicogenomics inform risk assessment? *Toxicology and Applied Pharmacology*, 207(2, Suppl):592–598.
- Odunsi, K., Wollman, R., Ambrosone, C., Hutson, A., McCann, S., Tammela, J., Geisler, J., Miller, G., Sellers, T., Cliby, W., Qian, F., Keitz, B., Intengan, M., Lele, S., and Alderfer, J. (2005). Detection of epithelial ovarian cancer using

- <sup>1</sup>H-NMR-based metabonomics. *International Journal of Cancer*, 113(5):782–788.
- Osborne, J. and Costello, A. (2004). Sample size and subject to item ratio in principal components analysis. Available at: <http://pareonline.net/getvn.asp?v=9&n=11>. Last visited on 2011-11-24.
- Park, Y., Tison, J., Lek, S., Giraudel, J., Coste, M., and Delmas, F. (2006). Application of a self-organizing map to select representative species in multivariate analysis: A case study determining diatom distribution patterns across France. *Ecological Informatics*, 1(3):247–257.
- Pierens, G., Palframan, M., Tranter, C., Carroll, A., and Quinn, R. (2005). A robust clustering approach for NMR spectra of natural product extracts. *Magnetic Resonance in Chemistry*, 43(5):359–365.
- Plumb, R., Granger, J., Stumpf, C., Johnson, K., Smith, B., Gaultitz, S., Wilson, I., and Perez, J. (2005). A rapid screening approach to metabonomics using UPLC and oa-TOF mass spectrometry: application to age, gender and diurnal variation in normal/Zucker obese rats and black, white and nude mice. *Analyst*, 130(6):844–849.
- Polani, D. (1999). On the optimization of self-organizing maps by genetic algorithms. In Oja, E. and Kaski, S., editors, *Kohonen Maps*, pages 157–169. Elsevier, Amsterdam, The Netherlands.
- Pözlbauer, G. (2004). Survey and comparison of quality measures for self-organizing maps. In *WDA 2004*, pages 67–82. Elfa Academic Press, Kosice. Vortrag: Workshop on Data Analysis, Vysoke Tatry, Slovakia, 2004-06-24 – 2004-06-27.
- Pugliatti, M., Beghi, E., Forsgren, L., Ekman, M., and Sobocki, P. (2007). Estimating the cost of epilepsy in Europe: A review with economic modeling. *Epilepsia*, 48(12):2224–2233.
- Ripley, B. (1987). *Stochastic Simulation*. John Wiley and Sons, New York, USA, second edition.
- Ross, A., Schlotterbeck, G., Dieterle, F., and Senn, H. (2007). NMR spectroscopy techniques for application to metabonomics. In Lindon, J., Nicholson, J., and Holmes, E., editors, *The Handbook of Metabonomics and Metabolomics*, chapter 3, pages 55–112. Elsevier, Amsterdam, The Netherlands.

- Rousseeuw, P. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20(1):53–65.
- Rubingh, C., Bijlsma, S., Derks, E., Bobeldijk, I., Verheij, E., Kochbar, S., and Smilde, A. (2006). Assessing the performance of statistical validation tools for megavariate metabolomics data. *Metabolomics*, 2(2):53–61.
- Ryan, D. and Robards, K. (2006). Metabolomics: The greatest omics of them all? *Analytical Chemistry*, 78(23):7954–7958.
- Salas, R., Moreno, S., Allende, H., and Moraga, C. (2007). A robust and flexible model of hierarchical self-organizing maps for non-stationary environments. *Neurocomputing*, 70(16-18):2744–2757.
- Sammon, J. (1969). A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers*, C-18(5):401–409.
- Schmidt, C. (2002). Toxicogenomics - an emerging discipline. *Environmental Health Perspectives*, 110(12):A750–A755.
- Seltmann, G., Voigt, W., and Beer, W. (1994). Application of physico-chemical typing methods for the epidemiological analysis of salmonella enteritidis strains of phage type 25/17. *Epidemiology and Infection*, 113(3):411–424.
- Sharaf, M., Illman, D., and Kowalski, B. (1986). *Chemometrics*, volume 82 of *Monographs on Analytical Chemistry and its Applications*. John Wiley & Sons, New York, USA.
- Sheskin, D. (2000). *Handbook of Parametric and Nonparametric Statistical Procedures*. Chapman & Hall/CRC, Boca Raton, Florida, USA, second edition.
- Silva, B. and Marques, N. (2007). A hybrid parallel SOM algorithm for large maps in data-mining. Available at: <http://ssdi.di.fct.unl.pt/~nmm/MyPapers/SM2007.pdf>. Last visited on 2012-06-27.
- Siuzdak, G. and Trauger, S. (2007). What is mass spectrometry? Technical report, Scripps Center for Mass Spectrometry.
- Solanky, K., Bailey, N., Hall, B., Bingham, S., Davis, A., Holmes, E., Nicholson, J., and Cassidy, A. (2005). Biofluid  $^1\text{H}$  NMR-based metabonomic techniques in nutrition research - metabolic effects of dietary isoflavones in humans. *Journal of Nutritional Biochemistry*, 16(4):236–244.

- Spraul, M., Neidig, P., Klauck, U., Kessler, P., Holmes, E., Nicholson, J., Sweatman, B., Salman, S., Farrant, R., Rahr, E., Beddell, C., and Lindon, J. (1994). Automatic reduction of NMR spectroscopic data for statistical and pattern recognition classification of samples. *Journal of Pharmaceutical & Biomedical Analysis*, 12(10):1215–1225.
- Sun, C., Teng, Y., Li, G., Yoshioka, S., Yokota, J., Miyamura, M., Fang, H., and Zhang, Y. (2010). Metabonomics study of the protective effects of *Lonicera japonica* extract on acute liver injury in dimethylnitrosamine treated rats. *Journal of Pharmaceutical and Biomedical Analysis*, 53(1):98–102.
- Suna, T., Salminen, A., Soininen, P., Laatikainen, R., Ingman, P., Mäkelä, S., Savolainen, M., Hannuksela, M., Jauhiainen, M., Taskinen, M., Kaski, K., and Ala-Korpela, M. (2007).  $^1\text{H}$  NMR metabonomics of plasma lipoprotein subclasses: elucidation of metabolic clustering by self-organizing maps. *NMR in Biomedicine*, 20(7):658–672.
- Sussulini, A., Prando, A., Maretto, D., Poppi, R., Tasic, L., Banzato, C., and Arruda, M. (2009). Metabolic profiling of human blood serum from treated patients with bipolar disorder employing  $^1\text{H}$  NMR spectroscopy and chemometrics. *Analytical Chemistry*, 81(23):9755–9763.
- Tan, H. and George, S. (2004). Investigating learning parameters in a standard 2-d SOM model to select good maps and avoid poor ones. In Webb, G. I. and Yu, X., editors, *AI 2004: Advances in Artificial Intelligence*, Lecture Notes in Computer Science, pages 425–437, Heidelberg, Berlin. Springer. 17th Australasian Joint Conference on Artificial Intelligence, Cairns, Australia, December 4–6, 2004.
- Taner, M. (1997). Kohonen’s self organizing maps with conscience. Technical report, Rock Solid Images.
- Tang, H., Xiao, C., and Wang, Y. (2009). Important roles of the hyphenated HPLC-DAD-MS-SPE-NMR technique in metabonomics. *Magnetic Resonance in Chemistry*, 47(2):S157–S162.
- Theodoridis, S. and Koutroumbas, K. (2003). *Pattern Recognition*. Academic Press, Elsevier, San Diego, USA, second edition.
- Van Bramer, S. (1998). An introduction to mass spectrometry. Technical report, Widener University.

- Varmuza, K. and Filzmoser, P. (2009). *Introduction to Multivariate Statistical Analysis in Chemometrics*. CRC Press, Boca Raton, Florida, USA.
- Velicer, W. (1976). Determining the number of components from the matrix of partial correlations. *Psychometrika*, 41(3):321–327.
- Vesanto, J. (1999). SOM-based data visualization methods. *Intelligent Data Analysis*, 3(2):111–126.
- Vesanto, J., Himberg, J., Alhoniemi, E., and Parhankangas, J. (2000). SOM toolbox for Matlab 5. Technical report A57, Helsinki University of Technology, Espoo, Finland.
- Villmann, T., Der, R., Herrmann, M., and Martinetz, T. (1997). Topology preservation in self-organizing feature maps: Exact definition and measurement. *IEEE Transactions on Neural Networks*, 8(2):256–266.
- Wang, D., Resson, H., Musavi, M., and Domnisoru, C. (2002). Double self-organizing maps to cluster gene expression data. In Verleysen, M., editor, *ESSAN 2002*, pages 45–50. 10th European Symposium on Artificial Neural Networks, Bruges, Belgium, 24–26 April 2002.
- Wang, Y., Yang, C., Mathee, K., and Narasimhan, G. (2005). Clustering using adaptive self-organizing maps (ASOM) and applications. In Sunderam, V., van Albada, G. D., Sloot, P., and Dongarra, J., editors, *Computational Science - ICCS 2005*, Lecture Notes in Computer Science, pages 944–951, Heidelberg, Berlin. Springer. 5th International Conference, Atlanta, May 22–25, 2005.
- Want, E., Nordstrom, A., Morita, H., and Siuzdak, G. (2007). From exogenous to endogenous: The inevitable imprint of mass spectrometry in metabolomics. *Journal of Proteome Research*, 6(2):459–468.
- Want, E., O’Maille, G., Smith, C., Brandon, T., Uritboonthai, W., Qin, C., Trauger, S., and Siuzdak, G. (2006). Solvent-dependent metabolite distribution, clustering, and protein extraction for serum profiling with mass spectrometry. *Analytical Chemistry*, 78(3):743–752.
- Webb, A. (2002). *Statistical Pattern Recognition*. John Wiley and Sons, West Sussex, England, second edition.
- Webb, K., Bristow, T., Sargent, M., and Stein, B. (2004). Methodology for accurate mass measurement of small molecules. Best practice guide, LGC, Teddington, UK.



- Weckwerth, W. and Fiehn, O. (2002). Can we discover novel pathways using metabolomic analysis? *Current Opinion in Biotechnology*, 13(2):156–160.
- Weckwerth, W. and Morgenthal, K. (2005). Metabolomics: from pattern recognition to biological interpretation. *Drug Discovery Today: Targets*, 10(22):1551–1558.
- Wickelmaier, F. (2003). An introduction to MDS. Technical report, Sound Quality Research Unit, Aalborg University, Denmark.
- Williams, C. (2002). On a connection between kernel PCA and metric multidimensional scaling. *Machine Learning*, 46(1-3):11–19.
- Williams, D. and Fleming, I. (1995). *Spectroscopic Methods in Organic Chemistry*. McGraw-Hill, Berkshire, England, fifth edition.
- Wilson, I., Nicholson, J., Perez, J., Granger, J., Johnson, K., Smith, B., and Plumb, R. (2005). High resolution ultra performance liquid chromatography coupled to oa-TOF mass spectrometry as a tool for differential metabolic pathway profiling in functional genomic studies. *Journal of Proteome Research*, 4(2):591–598.
- Xi, Y. and Roche, D. (2008). Baseline correction for NMR spectroscopic metabolomics data analysis. *BMC Bioinformatics*, 9(324):1–10.
- Yap, I., Clayton, T., Tang, H., Everett, J., Hanton, G., Provost, J., Net, J., Charuel, C., Lindon, J., and Nicholson, J. (2006). An integrated metabonomic approach to describe temporal metabolic dysregulation induced in the rat by the model hepatotoxin allyl formate. *Journal of Proteome Research*, 5(10):2675–2684.
- Yde, C., Bertram, H., and Knudsen, K. (2010). NMR-based metabonomics reveals distinct metabolic profiles of plasma from sows after consumption of diets with contrasting dietary fibre levels and composition. *Livestock Science*, 133(1-3):26–29.
- Zhu, C., Liang, Q.-L., Wang, Y., and Luo, G. (2010). Integrated development of metabonomics and its new progress. *Chinese Journal of Analytical Chemistry*, 38(7):1060–1068.
- Zuppi, C., Messana, I., Forni, F., Rossi, C., Pennachietti, L., Ferrari, F., and Giardina, B. (1997).  $^1\text{H}$  NMR spectra of normal urines: reference ranges of the major metabolites. *Clinical Chimica Acta*, 265(1):85–97.

- Zweiri, M. A., Sills, G., Leach, J., Brodie, M., Robertson, C., Watson, D., and Parkinson, J. (2010). Response to drug treatment in newly diagnosed epilepsy: A pilot study of  $^1\text{H}$  NMR- and MS-based metabonomics analysis. *Epilepsy Research*, 88(2-3):189–195.