Multivariate Consensus Trees:

Tree-based clustering and profiling for mixed data types

Thesis submitted by

Timothy Peter Hancock BSc(Hons)

in 2006

STATEMENT ON SOURCES

Declaration

I declare that this thesis is my own work and has not been submitted in any form for another degree or diploma at any university or other institution of tertiary education. Information derived from the published or unpublished work of others has been acknowledged in the text and a list of references is given.

…………………………………………….. ………………………

(Signature) (Date)

STATEMENT OF ACCESS

I, the undersigned, author of this work, understand that James Cook University will make this thesis available for use within the University Library and, via the Australian Digital Theses network, for use elsewhere. I understand that, as an unpublished work, a thesis has significant protection under the Copyright Act and;

I do not wish to place any further restriction on access to this work.

_____________________ ______________
Signature Date

ELECTRONIC COPY

I, the undersigned, the author of this work, declare that the electronic copy of this thesis provided to the James Cook University Library is an accurate copy of the print thesis submitted, within the limits of the technology available.

_______________________ ________________

Signature Date

Statement of Contributions

I would like to thank the following people and organizations for the funding they gave to me over the last three and a half years.

1. The Australian government postgraduate award (APA) scholarship.
2. DRS grant from James Cook University.
3. The department of Mathematics, Statistics, Physics, Computer Science and IT at James Cook University.
4. The department of Analytical Chemistry and Pharmaceutical Technology at the Vrije Univeriteit Brussels.

Acknowledgements

Studying the same thing for three and a half years and then writing a thesis to prove it attains a PhD. With this in mind I would first like to thank the people who have distracted me and made me realise that there is far more to life than tree-based models for mixed type clustering and profiling. In particular I would like to thank all mu friends and colleagues that have had to bare the load of everything that went even just a little bit wrong over the last three and a half years. My family in particular must also be mentioned as over the course of my PhD they had to put up with even more. Thanks Dad for reading mine opus. I would like to thank my supervisors, Danny Coomans and Bruce Litow, for the time and effort they gave such that my PhD was. Also I would like to thank the people at FABI for food, shelter and support for my trip over to Belgium, Europe and the world.

In short this thesis is the culmination of a lot of people's insight and effort that has been condensed into the next 200 odd pages, thanks all …

Abstract

Multivariate profiling aims to find groups in a response dataset that are described by relationships with another. Profiling is not predicting each variable within the response set, but finding stable relationships between the two datasets that define common groups. Profiling styles of analysis arise commonly within the context of survey, experimental design and diagnosis type of studies. These studies produce complex multivariate datasets that contain mixed variables often with missing values that require analysis with a flexible, stable statistical technique.

The profiling model under consideration within this thesis is a Classification and Regression Tree (CART). A standard CART model finds groups within a univariate response by building a decision tree from a set of predictor variables. The flexible structure of a CART model allow it to be used for either discriminate or regression analysis whilst also catering for mixed types within the predictor set.

The goal of this thesis to develop methods that extend CART for a multivariate response dataset involving mixed data types. Multivariate regression for CART (MRT) has recently been shown to be a powerful profiling and clustering tool. However the same successes in extending CART for multivariate classification and multivariate mixed type analysis is yet to be realised. To begin with thesis explores simple extensions to CART for multivariate mixed type analysis. These are binary substitution of categorical variables within the response set and partitioning of a distance matrix using Db-MRT. These techniques use already existing extensions to

CART methods and are used as comparison methods to gauge the performance of the ensemble and consensus approaches that are the focus of this thesis.

Ensemble models using CART, such as random forests and treeboost, not only improve the overall accuracy of the model predictions but also introduce an ensemble proximity matrix as a measure of similarity between observations of the response set. In this thesis, through MRT, extensions to both random forests and treeboost are developed such that they predict a multivariate response. Furthermore, by binary substitution of the categorical variables within the response set these multivariate ensemble techniques are further extended to mixed type profiling. A result of this extension is that the ensemble proximity matrix now describes the groups found within the multivariate response. In this way multivariate tree-base ensembles can be interpreted as a cluster ensemble method, where the ensemble proximity matrices can be seen as cluster ensemble consensus matrices. In this thesis these proximity matrices are found to be powerful visualisation tools providing improved resolution of group structure found by a multivariate ensemble method. More so, as in cluster ensembles using these matrices as an input in to a clustering method improves the accuracy of the groups found.

The main work of this thesis is the development of the Multivariate Consensus Tree (MCT) framework for mixed type profiling. Motivating the MCT approach is the need to further understand which variables relate to the groups observed within the proximity matrix. To do this MCTs describe three methods to intelligently combine the ensemble proximity matrices of individual responses into one overall consensus matrix. This consensus matrix is a summary of the overall group structure within each individual proximity matrix. As MCTs work solely with proximity matrices they are independent of the data types within the variables of the response set. Furthermore as each response variable is explicitly predicted it is possible to assess the quality of each proximity matrix in terms of predictive accuracy of the corresponding ensemble.

The MCT consensus matrix is a visualisation tool for the groups present within both the response and predictor datasets. As a consensus matrix is a similarity matrix this thesis proposes five new splitting criteria for tree-based models that search for decision rules within variables of the predictor set that partition the consensus matrix into the observed groups. This tree provides a logical decision path that predicts each group. As the groups within the response are now defined by their relationships within the predictor set, the MCT profiling is complete. This thesis proposes two algorithms for building an MCT; global MCTs and local MCTs. Global MCTs construct an overall consensus matrix spanning all observations, and recursively partition on this matrix to build the tree. Local MCTs build a new consensus matrix at each terminal node to evaluate each new split.

As MCTs have the proximity matrices to summarise the group structure within each response variable methods to identify important subgroups within these variables are also proposed. This search for subgroups within the response can be done on two levels. Firstly to identify subgroups of response variables for overall analysis; and secondly to identify subsets of response variables within any specific group found by the MCT. By finding subsets of response variables that relate to specific group structure the understanding of structure within the dataset is greatly improved.

This thesis shows tree-based methods for profiling, in particular MCTs, to be a powerful tool for mixed type analysis. Firstly, the visualisation of the tree, combined with the proximity matrices, provide a unique view of the groups found and allow for their easy interpretation within the context of the analysis. Secondly, MCTs are shown to accurately estimate the number of groups and provide measures on their stability and accuracy. Furthermore, MCTs are found to be resistant to noise variables within the analysis. Finally they provide methods to find subgroups within the response variables and to identify unimportant variables from the analysis. Throughout this thesis these tree-based methods are compared with standard clustering techniques to provide an accurate benchmark for their performance.

Table of Contents

List of Figures

List of Tables

1. Introduction

One of the functions of statistics is to find and describe important features within a dataset. In most cases, what the statistics finds is expected. But as the questions asked become more complicated and involve identifying interactions across many variables of different types, even the experts cannot describe all the effects that combine to produce the outcome. Researchers in an assortment of fields such as ecology, psychology, bioinformatics and medical research are now gathering highly complex datasets that require large amounts of detailed analysis to comprehend. So much so that the complexity of these datasets is driving the development of new approaches that are powerful enough to analyse them whilst also being easily understood. This is where the full potential of statistics as a data mining tool is realised.

Multivariate profiling aims to find groups that exist over many variables. Profiling is not predicting each variable, but finding stable groups that exist over all variables. Cluster analysis can be considered a subset of profiling called "*unconstrained profiling*", where there is only a single dataset. However profiling can be extended to finding groups in one dataset that are well represented within another. This is "*constrained profiling*" where the structure found in the response set is constrained by the relationships available within the predictor set. Profiling analyses arise most commonly within the context of survey, experimental design and diagnostic styles of analyses.

Most of the complexity of clustering and profiling analysis is in the validation of the model. The groups found must be shown to be logical and be representative of the data. **This thesis explores the potential of Classification and Regression Tree (CART) models for use in multivariate profiling problems.** CART is an ideal choice for profiling as it provides an intuitive framework for understanding relationships within a dataset. CART models use a hierarchy of decision rules found on predictor variables that logically determine groups within response variables.

A Classification and Regression Tree (CART) (Breiman, Friedman, Olshen and Stone 1984) is a data-mining tool for non-linear regression and classification. CART works by creating a binary tree from the set of predictor variables and by imposing conditions upon these variables, the tree predicts or classifies a response. The resulting tree provides information on the relationships between predictor variables and the response, and gives an insight into possible groups or clusters within the dataset. CART modelling provides an intuitive statistical framework that is easily understood by the non-statistician. Due to the ease of interpretation many researchers are now choosing to use CART models over the standard regression or classification techniques (Quinlan 1986, De'ath and Fabricius 2000, E. Dusseldorp and J. J. Meulman 2001).

By creating an ensemble of CART models it is well established that predictive performance will stabilise and improve (Dietterich 2000b, Breiman 2001). Another motivating reason for this is to extract ensemble co-occurence matrix of the forest, called the Random Forest Proximity (RF proximity) matrix (Breiman 2001, Shi and Horvath 2006). Contained within this RF proximity matrix is a summary of all the group structure within the response set as defined by the predictor variables. This is viewing tree-base ensembles as a cluster ensemble method (Strehl and Ghosh 2002), where the ensemble proximity matrices can be seen as cluster co-occurrence matrices (Monti, Tamayo, Mesirov and Golub 2003). These matrices are powerful visualisation tools providing improved resolution of group structure found by an ensemble method.

Extending CART to multivariate profiling is finding appropriate measures to assess the quality of a split over many variables of different types. For multivariate regression this can be easily implemented by using multivariate sums of squares, Multivariate Regression Trees (MRT) (Segal 1992). However such simple extensions are not possible for multivariate classification or mixed type response sets. A generalized entropy approach for multiple binary responses (Zhang 1998) uses a log-linear model over the responses but assumes an exponential distribution for each variable for each terminal node of the tree. General estimating equations have also been used to extend trees for a mixed type response set (Seong Keon Lee, Hyun-Cheol Kang, Sang-Tae Han and Kwang-Hwan Kim 2005). This approach uses a marginal regression model to determine the terminal nodes of the tree.

As model based methods assume a distribution of the response variables, they remove the non-parametric nature of CART. As such, these methods view multivariate extensions to CART in a multivariate predictive framework. This thesis takes a different view, choosing to use multivariate trees as a method for identifying stable groups within the datasets. These two ideas are not same as finding a stable group structure may not lead to optimal predictive performance.

Partitioning a distance matrix as in Db-MRT (De'ath 2002) offers a non-parametric approach for extending CART to multivariate response sets. Although transforming the response into a distance matrix allows for an easy implementation of multivariate

CART, uncertainty exists in the measuring of the quality of the tree's predictions and in determining the size of the tree. Furthermore, standard distance measures are poorly defined over a mixed type domain. Methods that can give an indication on the quality of a node in a mixed multivariate domain are required for a complete solution.

Taking lead from DB-MRT and tree-based ensemble methods this thesis investigates using the RF proximity to relating observations as input to a clustering or profiling model. To identify the group structure within the RF proximity matrix we propose a Multivariate Consensus Tree (MCT) for partitioning the matrix to identify groups within the response variables. A MCT searches for decision rules within the predictor variables that define areas of high similarity within the RF proximity matrix.

One important issue with cluster or profiling type analyses is that different data types may exist within the variables of the datasets involved. If presented with a mixed type dataset the question of an appropriate way of relating objects spanning many types must be answered. A key feature of the base CART model is its power in handling mixed type datasets. In this thesis the suitability of tree-based models to solve such a problem is explored.

By using CART and MRT theory as a starting point this thesis also proposes and compares methods to create RF proximities over a mixed type dataset. The first proposed method substitutes the categorical variables with a binary indicator matrix (Gifi 1990). From here the substituted response set is treated as a multivariate regression problem and a tree based ensemble is built. Then the RF proximity matrix is extracted and an MCT is used to identify the groups.

However employing binary substitution on the categorical responses assumes a Euclidean relationship between categorical and continuous variables and homogenous levels within each categorical variable (Kaufman and Rousseeuw 1990). It is expected that for a realistic analysis this assumption may not be valid.

Motivated by the need to further understand which variables relate to the groups observed within the proximity matrix, the second mixed type extension proposed is to intelligently combine individual variable RF proximity matrices to produce a single overall consensus matrix. To do this the three combination methods are explored, General Procrustes Analysis (GPA), a Beta Binomial model (Gelman, Carlin, Stern and Rubin 1997) and Plaid Models (Lazzeroni and Owen 2002).

As this approach models a variable's proximity matrix it is independent of its data type, therefore allowing it to produce a consensus matrix from proximities created from a mixed type data set. Furthermore as the individual variable proximity matrices are predicted by the overall consensus a R^2 can be defined as a measure of variable importance is to the final cluster solution can be computed. A MCT is then used to identify the groups within the overall consensus based on decision rules found within the original dataset variables.

In Section 2 the background literature for clustering and profiling with a particular focus on ensemble and tree-based methods is reviewed. Section 3 describes the specifics of the core methods required to understand MCTs including univariate and multivariate tree models, tree-based ensembles, mixed type extensions to trees and methods to combine proximity matrices are described. Section 4 presents the MCT algorithms using the iris dataset as an example and Section 5 references the software used and developed in this thesis. Sections 6 and 7 use simulated and benchmark examples to assess and compare the performance of individual trees, tree-ensembles for profiling and MCT methods. Section 8 and 9 present a detailed discussion on the performance of the methods and conclusion chapters.

2. Background

2.1 Relating Objects

Measures of how related objects are within a dataset forms one of the core principles of statistics. How objects are related depends on their data types. Two broad statistical groups of variable types exist: quantitative and qualitative. Quantitative or ‘continuous’ variables comprise of two subgroups, interval and ratio. These variables have a strict order and are commonly used to measure the relative magnitude of an observation, e.g. temperature. Qualitative or ‘categorical’ variables also have two subgroups: ordinal and nominal. However nominal qualitative variables can be unordered and their assigned labels may not be representative of the actual levels within the variable. Categorical variables are often used to identify grouping structure over a variable e.g. hair colour. Due to the structural difference between the variables, separate measures for relating objects within a single data type have been developed.

For relating objects based on quantitative variables the most common choice is the Euclidean distance (Everitt 1993). The Euclidean distance between two vectors $\underset{\sim}{x}$ and $\underset{\sim}{y}$ is defined as,

$$d\left(\underset{\sim}{x},\underset{\sim}{y}\right)=\sqrt{\left(\underset{\sim}{x}-\underset{\sim}{y}\right)^{T}\left(\underset{\sim}{x}-\underset{\sim}{y}\right)}\ . \tag{2.1}$$

The interpretation of (2.1) is the length of a straight line between the two vectors and therefore assumes a continuous domain. As the length of the line increases the less related the two observations are and therefore the Euclidean distance is also a dissimilarity measure.

Common choices for relating objects based on qualitative variables are a simple pairwise matching statistic (Kaufman, et al. 1990) that counts the number of observations over the variables with the same labels or using the chi-square distance (Lebart, Morneau and Warwick 1984). These are similarity measures as the more related the two objects are the larger the value of the measure.

2.2 Relating Mixed Types

Mixed type analysis relates variables spanning multiple data types. The problem surrounding this is the order inherent in quantitative variables and the lack of order of qualitative variables. This structural difference between the two types prevents any simple extension of a measure designed for a single type. There are however, approaches commonly used for relating mixed types. Two of these approaches are discussed in this thesis and are used as benchmark methods. These are using the Gower dissimilarity measure and binary substitution of categorical variables.

The first benchmark method for relating mixed types is the *Gower dissimilarity measure* (Gower 1971a). This is defined as,

$$d\left(\underset{\sim}{x}_i, \underset{\sim}{x}_j\right) = \frac{\sum_{m=1}^{M} \delta_{ij}^{(m)} d_{ij}^{(m)}}{\sum_{m=1}^{M} \delta_{ij}^{(m)}} \tag{2.2}$$

where $d_{ij}^{(m)}$ is the distance between the observations *i* and *j* and carries a different definition given the type of variable *m*, and $\delta_{ij}^{(m)}$ is a binary flag indicating the position of missing values within a variable. The definitions of $d_{ij}^{(m)}$ are:

1. For nominal variables:

$$\begin{aligned} d_{ij}^{(m)} &= 1 \text{ if } x_{im} \neq x_{jm} \\ &= 0 \text{ if } x_{im} = x_{jm} \end{aligned} \qquad (2.3)$$

2. For binary variables, $d_{ij}^{(m)}$ is the Jaccard Coefficient (Kaufman, et al. 1990).

3. For interval or ratio variables:

$$d_{ij}^{(m)} = \frac{\left| x_{im} - x_{jm} \right|}{range(x_m)} \qquad (2.4)$$

This dissimilarity provides a simple measure between mixed types, however assigns the same weight to any variable type and therefore will not model complex relationships within the data.

Binary variable substitution for categorical variables is the second method for relating mixed types. This method substitutes categorical variables into an indicator matrix *G* (Young 1981, Gifi 1990). An example of binary substitution of a categorical variable (2.5) shows the indicator matrix G has a ‘1’ at the location of each category in the categorical variable $x_{categorical}$:

$$x_{categorical} = \left[1,1,1,2,2,2,3,3,3\right] = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} = G \qquad (2.5)$$

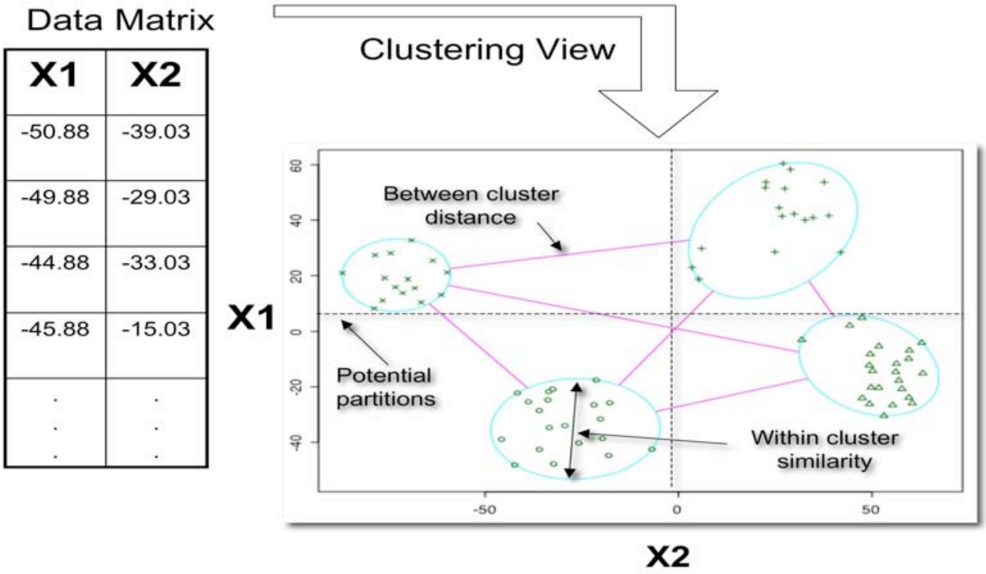
This substitution is made in optimal scaling, non-linear principal component analysis and correspondence analysis (Lebart, et al. 1984).

Both the Gower dissimilarity and binary substitution assume Euclidean distances between categorical and continuous variables (Buuren and Heiser 1989). This assumption has considerable impact on the structure within the qualitative variables. It forces the assumption that the levels within these variables are homogeneous or evenly spaced (Kaufman, et al. 1990). Although, to identify a simple group structure this assumption may be valid, it is not likely to hold for more complex relationships. Therefore to find group structure over more complex relationships spanning multiple types a more advanced method for relating objects is required.

2.3 Cluster Analysis

The aim of cluster analysis is to find groups within a single dataset. There are many clustering methods available. Cluster analysis methods differ either in their definition of a group or in the algorithm used for finding the groups. Common group definitions of are either based on within group measures such as areas of high similarity between observations or between group measures such as the maximum distance between two objects (Figure 1).

Figure 1: Cluster analysis example.



X1	X2
-50.88	-39.03
-49.88	-29.03
-44.88	-33.03
-45.88	-15.03
. . .	. . .

There are many algorithms commonly used for cluster analysis. Firstly, there are divisive or agglomerative methods, which iteratively separate or merge objects into groups. There are a variety of these methods such as hierarchical agglomeration (Everitt 1993) and auto-associative multivariate regression trees (AA-MRT) (Questier, Put, Coomans, Walczak and Vander Heyden 2004). Secondly there are optimisation methods that search for a predefined number of stable group centres. These methods define clusters by minimising the distance between objects and the group centres. K-means and partitioning around medoids (PAM) (Kaufman and Rousseeuw 1987) are examples of optimisation based methods.

2.3.1 Hierarchical agglomeration

Hierarchical agglomeration iteratively merges the objects into groups starting with all objects in their own group or each object by itself. Using a specified agglomeration or merging criterion the algorithm searches for two objects to form the next best group. Once a group of more than one object is formed, these objects are considered a single group and treated as one for the duration of the algorithm. As the algorithm progresses all objects are placed into the group to which they are most similar, based on the agglomeration criterion. These groups are iteratively merged until the final group is the entire dataset. At any point in the agglomeration the algorithm may be stopped and the individual groups at that iteration can be extracted.

Different merging criteria will find different groups. Commonly used criteria are:

1. Single Linkage: The distance between two groups is the minimum distance between any two objects in separate groups.
2. Complete Linkage: The distance between two groups is the maximum distance between any two objects in separate groups.
3. Average Linkage: The distance between two groups is the average distance between all objects in both groups.
4. Wards Method: The distance between groups is defined by the change in within sums of squares between the merged and unmerged groups.

Determining where to stop the merging determines the number of clusters that have been found. A plot of the agglomeration history called a ‘dendrogram’, for a small dataset can help to determine the stopping location. For large datasets however, these are difficult to read and no accepted automatic stopping criteria are available (Milligan and Cooper 1985).

2.3.2 K-means and medoids (PAM)

K-Means (Hartigan 1975) searches for the optimal set of clusters that minimise their within cluster sums of squares (WSS),

$$WSS = \sum_{k=1}^{K} \sum_{x_i \in C_k} \left(\underset{\sim}{x}_i - \overline{\underset{\sim}{x}}_k \right)^2 \tag{2.6}$$

where $\underset{\sim}{x}_i$ is an observation vector in the dataset, C_k are the labels of the set of K clusters $\{1, \cdots, k, \cdots, K\}$ and $\overline{\underset{\sim}{x}}_k$ is the mean vector of cluster *k*. The user specifies the number of clusters to find and the algorithm starts with a randomly generated set of *K* cluster centres. Each object is then assigned to the centre it is closest to. After the assignment the cluster centres are re-computed and the objects re-assigned. The algorithm is stopped after the cluster centres have stabilised. K-Means requires quantitative variables as inputs.

K-medoids is a robust form of K-Means as instead of minimising the within cluster sums of squares, it searches for representative objects within the dataset to form the cluster centres. These centres are selected such that the absolute distance between them is maximised. Partitioning Around Medoids (PAM) (Kaufman, et al. 1987) is an implementation of K-medoids. PAM is more robust than K-Means as the absolute distance between cluster centres is less affected by outlying observations than the squared distance.

2.3.3 Clustering challenges

Cluster analysis is an unsupervised search for grouping structure within a dataset. However, as different groups are found using different techniques, verifying the accuracy of any clustering solution is difficult. The problem is a lack of a known objective to compare the final result against. This manifests itself in two major issues for any cluster analysis technique:

1. Estimating the number of groups to be found.

2. Verifying the accuracy, stability and reproducibility of these groups.

These problems can never be fully resolved by any method. Despite this much research has been conducted with the aim to assist those implementing clustering analysis methods in validating their results.

2.3.4 Determining the number of groups

The first studies into determining the number of groups in a dataset focus on automatic stopping rules for hierarchical agglomeration techniques. A stopping rule dictates where to stop merging objects to determine the number of groups found by the scheme. Comprehensive simulation tests of 30 of these criteria (Milligan, et al. 1985) revealed a clear best set of indices but also a wide variety of performances, and concludes unsurprisingly that the performance of each criterion is dataset dependent.

Predictive arguments for determining the number of clusters in a dataset are becoming more popular, as they can be explained in terms of model complexity. The elbow of the relative error curve of an Auto-associative Multivariate Regression Trees (AA-MRTs) (Smyth, Coomans, Everingham and Hancock 2005) is an example of model complexity determining cluster number. Predictive cluster number determination treats cluster analysis as optimising the performance of a multivariate predictive model.

Mixture model based methods (Fraley and Raftery 2002) estimate the data using a weighted sum of distributions where each distribution corresponds to a group. Mixture models are predictive models of the data and as such use model complexity measures such as the Bayesian Information Criterion (BIC) and likelihood ratio statistics to determine the number of clusters. Although these measures are seen as a useful tool in practice, they assume a known structure on the data that is not nessarily valid (Mclachlan, Bean and Peel 2002).

The Gap statistic (Tibshirani, Walther and Hastie 2001) estimates the number of clusters by searching for the most reproducible set of labels. The Gap statistic,

$$Gap_n(k) = E_n^*(\log(W_k)) - \log(W_k) \tag{2.7}$$

observes the change in the log sum of the pairwise distances, $\log(W_k)$, for all objects in each cluster *k* on the real dataset, compared to the mean of the log sum of pairwise distances for cluster *k* over many simulated reference datasets $E_n^*(\log(W_k))$. The gap statistic's use of simulated reference datasets provides a null distribution upon which to compare a potential clustering solution against. The optimal number of clusters is found when the pairwise distances between objects in the actual scheme are most different to pairwise distances in the reference datasets. This is at the maximum of the gap statistic and the most reproducible number of groups.

Figure-of-merits (FOM) (Yeung, Haynor and Ruzzo 2001) are also based on the idea that model stability rather than pure predictive performance should determine the optimal number of groups. To determine stability, jack-knife cross-validation on the variables is employed. As the number of clusters is increased, the model stability is assessed by monitoring the change in the root mean squared error over the course of the jack-knifing. The result is a curve similar to the relative error curves produced by auto-associative multivariate regression trees (AA-MRT) however the focus is on stability not predictive performance.

The FOM approach is similar in ideology to that of Tibshirani et al. (2005) and (Dudoit and Fridlyand 2002) where V-fold cross-validation on the observations is used to assess cluster stability. Tibshirani et al. (2005) compares the performances of the same clustering model run on a test/training set partition of the data. The performance of the clustering algorithm is assessed by comparing the predicted groups using the training model on the test set, with the groups found by clustering the test set individually. If the groups differ then the clustering algorithm is not stable. Dudoit and Fridlyand (2002) perform a similar cross validation however build a classifier to predict the group labels of the training set. This classifier is then used to predict the test set labels. These methods test the reproducibility of a clustering scheme, where the most reproducible number of clusters is selected as optimal, they require no known group labels and unlike trees or mixture models are independent of the clustering method.

Once the number of clusters has been determined there is still the problem of assessing the accuracy of the solution. Even with the number of clusters estimated there are still a huge number of possible group configurations. The rand and modified rand indices (Hubert and Arabie 1985) are measures of overlap between two clustering schemes on the same data set. If two different clustering schemes on the one dataset find similar grouping structure, then it is likely that the found groups are representative of the dataset. This is a relative form of accuracy, however as the group labels are unknown it is the only form of accuracy possible.

One solution to the problem of determining which clustering algorithm to use is by combining information over many different clustering algorithms. Such an approach is called an ensemble approach to clustering and is now discussed in more detail.

2.3.5 Cluster ensembles

A cluster ensemble (Strehl, et al. 2002) is simply a collection of clustering solutions that are combined into one overall solution. An individual clustering solution is called a partition within the cluster ensemble. The goal of the analysis is to find common grouping structure across each partition and summarise it into one overall partition. This is done to remove the need to choose which clustering method to use. Instead, a range of techniques are selected and the ensemble combines them into an overall partition which is the final grouping structure of the data. It is hoped by combining information across many solutions that the stability of the final clustering structure is improved.

2.3.5.1 Cluster ensemble objective functions

To find the overall partition requires a means of summarising common information across many different partitions. Functions that do this are called ensemble objective functions. In fact there are many potential statistics that can be employed as an objective function for cluster ensembles. The goal of a cluster ensemble is to find an overall partition that optimises this objective function.

Cluster ensembles were initially defined using the maximum of the normalised mutual information index, ϕ^{NMI}, between two partitions (Strehl, et al. 2002, Ana L. N. Fred and Anil K. Jain 2005) as its objective function:

$$\phi^{(NMI)}\left(\lambda^{(a)},\lambda^{(b)}\right)=\frac{2}{n}\sum_{l=1}^{k^{(a)}}\sum_{h=1}^{k^{(b)}}n_l^{(h)}\log_{k^{(a)}.k^{(b)}}\left(\frac{n_l^{(h)}n}{n^{(h)}n_l}\right) \tag{2.8}$$

In (2.8) the two partitions $\lambda^{(a)}$ and $\lambda^{(b)}$ on *n* observations have $k^{(a)}$ and $k^{(b)}$ groups where $n^{(h)}$ is the number of observations is cluster *h* according to $\lambda^{(a)}$, n_l is the number of observations in cluster *l* according to $\lambda^{(b)}$ and $n_l^{(h)}$ is the number of observations in cluster *l* according to $\lambda^{(b)}$ that are also in cluster *h* according to $\lambda^{(a)}$.

Defining (2.8) as the ensemble objective function, to find the overall partition $\hat{\lambda}$, will require a search over permutations of all labels within each partition in the set of all partitions, $\Lambda=\left\{\lambda^{(1)},\cdots,\lambda^{(q)},\cdots,\lambda^{(r)}\right\}$. Unfortunately this requires approximately ${k^n}/{k!}$ comparisons between groups where *n* >> *k* which is infeasible if the number of partitions is large (Strehl 2002). To bypass this computational complexity, cluster ensembles use heuristics to approximate $\hat{\lambda}$.

Another objective function commonly used defines the Euclidean dissimilarity (Weingessel, Dimitriadou and Hornik 2001) between two partitions to be,

$$d\left(\lambda^{(a)},\lambda^{(b)}\right)=\min_{\Pi}\left\|\lambda^{(a)}-\lambda^{(b)}\Pi\right\| \tag{2.9}$$

where the minimum is taken over all possible permutations Π, and $\|\,.\,\|$ is the Frobenius norm. The minimum of (2.9) is found when the permutation performed on $\lambda^{(b)}$ matches best the group configuration in $\lambda^{(a)}$.

A solution to (2.9) is possible by treating the problem as a linear sum assignment problem and using a linear program to estimate the overall partition (Hornik 2006). By extending (2.9) to the squared Euclidean dissimilarity,

$$d\left(\lambda^{(a)},\lambda^{(b)}\right)=\min\left\{\left(\lambda^{(a)}-\lambda^{(b)}\Pi\right)^{2}\right\} \tag{2.10}$$

another possible solution can be found using the iterative approach described in the "Voting" algorithm (Weingessel, et al. 2001). This algorithm iteratively updates the probability that each observation lies within each group over all partitions within the ensemble. The group with the maximum probability is the final clustering assignment.

Hypergraph representations of the ensemble as in Hypergraph Partitioning Algorithm HGPA, Meta-Clustering Algorithm MCLA (Strehl 2002) and Hybrid Bipartite Graph Formulation HBGF (Fern and Brodley 2004) also present a possible means for estimating the overall partition. These methods represent each cluster in the ensemble as a vertex on a graph that are connected by common observations. From this representation, a distance between clusters can be formulated, and the goal of the analysis is to collapse the graph into strongly connected components that define the overall partition.

Mixture model approaches can also be used to estimate the overall partition (Topchy, Jain and Punch 2004, Topchy, Minaei-Bidgoli, Jain and Punch 2004). Here each partition within the ensemble is treated as a random variable that can be modelled with a mixture of multivariate component densities,

$$P\left(\lambda_q \middle| \Theta\right) = \sum_{q=1}^{r} \alpha_q P\left(\lambda_q \middle| \theta_q\right) \tag{2.11}$$

where λ_q is a partition in the ensemble with component density parameters θ_q and α_q are the mixture weights and Θ is the set of all parameters of the mixture model to be estimated $\Theta = \left\{\alpha_1,\ldots,\alpha_r;\theta_1,\ldots,\theta_r\right\}$. The final cluster assignments are found by estimating the maximum posterior probability of each observation belonging to each component density in the mixture model.

2.3.5.2 Cluster ensemble consensus matrices

The most common solution to cluster ensembles avoids optimising an objective function entirely through the construction of a 'consensus' or 'co-occurrence' matrix over the observations. A consensus matrix is a similarity matrix where each cell contains a count as to how many times two observations are clustered together over all partitions in the ensemble. Once constructed from an ensemble this matrix is passed as an input to another clustering algorithm to find the overall partition. Examples of this approach can be found in the Cluster-based Similarity Partitioning Algorithm (CSPA) (Strehl, et al. 2002) and the evidence accumulation algorithms (Ana L. N. Fred, et al. 2005). CSPA clusters co-occurrence matrices using the hypergraph partitioning algorithm METIS (Karypis and Kumar 1998) and evidence accumulation using hierarchical agglomeration with single linkage (Section 2.3.1).

One approach to constructing a consensus matrix is by bootstrapping a clustering algorithm and computing the similarity based on the partitions from each bootstrapped model (Monti, et al. 2003). Bootstrapping for consensus construction has been applied to hierarchical agglomeration (Kavsek, Lavrac and Ferligoj 2001, Monti, et al. 2003, Stephen Swift, et al. 2004), PAM (Dudoit and Fridyland 2003) and self organising maps (SOM) (Monti, et al. 2003), and has consistently shown to improve the stability of the groups found. Furthermore, observation of the consensus matrix reordered by the found clusters provides a useful visualisation tool to assess the quality of the groups found (Ben-Hur, Elisseeff and Guyon 2002, Monti, et al. 2003).

Analysis of the structure within consensus matrices provides a method to estimate the optimal number of groups (Ben-Hur, et al. 2002, Monti, et al. 2003). This work assumes that an ideal consensus matrix is block diagonal and sparse. Therefore an ideal distribution for a consensus matrix can be estimated. This distribution can be viewed by a histogram, and should show two clear bins, one for the observations classified together and one for the observations not classified together. By computing the empirical cumulative distribution function (CDF) of the histogram and observing its structure a measure of quality for that scheme is produced. By observing changes in the CDFs of the same method grown to different numbers of clusters an estimate of the optimal number of clusters can be achieved.

Cluster ensembles show that by optimising the level of agreement between different clustering regimes, uncertainty in choosing the clustering method and estimating number of groups can be reduced. However these methods do not take into account to the accuracy of the clustering method. In fact a danger of these methods is that in combining many partitions without knowledge of how representative each solution is of the data it is possible to find a stable set of clusters with no accuracy. One solution to this problem is to consider each clustering algorithm as a prediction of the dataset.

By doing this ideas from predictive ensembles can be used to assess the accuracy of a cluster ensemble.

2.3.6 Cluster ensembles and predictive ensembles

By bootstrapping a clustering technique and aggregating the results, consensus clustering is essentially ‘bagging’ a clustering algorithm (Breiman 1996a). Usually bagging is a methodology that aims to optimise performance of a model by averaging over many bootstrapped predictions. In random forests it is shown that bagging tree-based models can dramatically improve their predictive performance (Breiman 2001). Furthermore, a random forest provides a means of summarising the structure found over all trees generated over the course of the bootstrapping through the construction of a proximity matrix. The random forest proximity matrix is similar to a consensus matrix, as it is a similarity matrix comprising of a count of how many times two observations have been placed in the same terminal node over an ensemble of trees within the forest. An unsupervised extension to random forests, (unsupervised random forests), has allowed for this matrix to be constructed on a single dataset by using a simulated response variable. Unsupervised random forests (Shi, et al. 2006) is a ensemble clustering algorithm that produces a consensus matrix and employs PAM on the proximity matrix to find the overall partition. This method exploits the advantages of consensus clustering, however as it uses a meaningless simulated response it nullifies the predictive improvements offered by random forests.

This thesis presents a new approach for cluster analysis of a mixed dataset called Multivariate Consensus Trees (MCT). MCTs exploit the similarity between the predictive proximity matrix of tree-based ensembles and the cluster ensemble consensus matrices. Combining the two ideas allow MCTs to harness the predictive power of tree-based ensembles, and by using cluster ensembles, to present this information in the form of a consensus clustering problem. Furthermore as trees are a model based clustering approach, MCTs also provide measures to estimate the optimal number of clusters and to assess the accuracy of the final solution.

MCTs predict each variable within the response set with a tree-based ensemble. From these ensembles the grouping structure from each variable is summarised into a proximity matrix. Then in a similar step to ensemble clustering these proximity matrices are combined into one overall consensus matrix. This consensus matrix provides an overall view of the group structure within the entire dataset. By searching for a decision within the original variables of the dataset, a tree is grown to partition this overall consensus matrix. The resulting tree is the called the MCT of the dataset and the groups found lie in the terminal nodes.

By predicting each variable individually MCTs perform a similar step to the cross validation used in FOMs, and allow for a way to assess how representative each variable is of the final clustering solution. This addresses the previously mentioned problem associated with consensus ensembles in assessing the accuracy of the final solution. Furthermore, knwledge of individual performances for each variable allows MCTs to perform a data reduction step to remove unimportant variables from the analysis. Variable selection performed in MCTs can reduce a highly complex clustering problem into a simpler one allowing for easier understanding of the final solution.

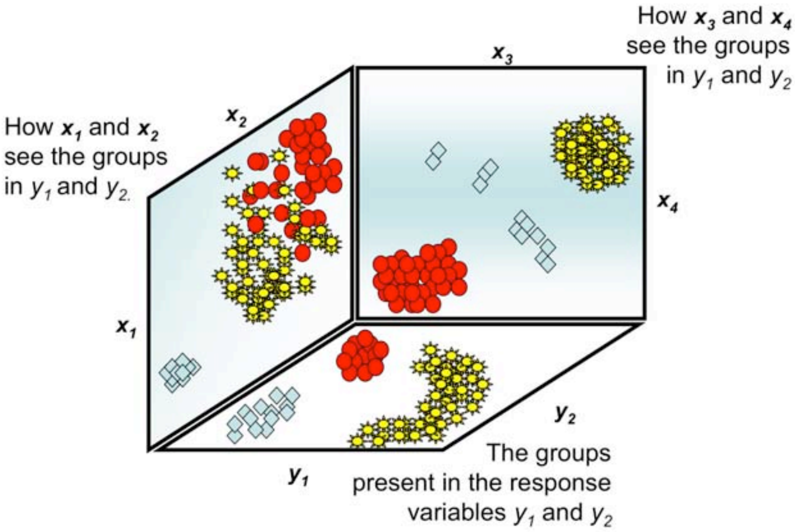
MCTs as a cluster analysis method use auto-association as in AA-MRT on the dataset to produce the tree. However as trees can predict a separate response dataset, MCTs can also be used as a multivariate profiling tool. In the next section common profiling methodology is discussed.

2.4 Profiling Analysis

The process of identifying groups in one dataset (the predictor set) that also define groups in another (the response set), is in this thesis is “*profiling the response dataset by the predictor set dataset*”. The response $Y = \left[\underset{\sim}{y}_1, \underset{\sim}{y}_2, \ldots, \underset{\sim}{y}_m, \ldots, \underset{\sim}{y}_M\right]$ consists of *n* observations on *M* variables where each response variable is denoted by $\underset{\sim}{y}_m$, and the predictor set $X = \left[\underset{\sim}{x}_1, \underset{\sim}{x}_2, \ldots, \underset{\sim}{x}_p, \ldots, \underset{\sim}{x}_P\right]$ on the same objects as the response set but on *P* separate predictor variables denoted as $\underset{\sim}{x}_p$. The response set is usually a small quite specific set of variables relating to the phenomena under consideration. However, it is common for the predictor set to consist of a comparatively large number variables that may be related to the response. In this case the researcher wishes to identify a small subset of predictor variables that summarise strong relationships.

Simply clustering the profiling set is an insufficient solution to profiling analysis, as the same grouping structure may not be present in the predictor set (Figure 2). Profiling methods require a compromise solution between the groups present within both the response and predictor datasets. If a group exists within the response set that is not represented within the predictor set then it will not be identified by a profiling method. Therefore profiling is a data mining problem as it involves identifying the important structure within the predictor variables that agrees with the structure of the response set. This requires a method that can not only relate objects within an individual dataset but also between datasets.

Figure 2: Example of profiling a response set $Y = \{y_1, y_2\}$ by predictor variables $X = \{x_1, x_2, x_3, x_4\}$.


How x_3 and x_4 see the groups in y_1 and y_2
x_3
How x_1 and x_2 see the groups in y_1 and y_2.
x_2
x_4
x_1
y_2
y_1
The groups present in the response variables y_1 and y_2

Relating two complete datasets is a difficult statistical problem and requires the specification of a statistical model. Methods like multivariate regression and Multivariate Analysis Of Variance (MANOVA) (Seber 1984), Canonical Correspondence Analysis (CCA) (Hotelling 1936), Generalised Procrustes Analysis (GPA) (Gower 1975) all relate information from many sources. These methods however are either aimed more at explicit prediction or summarising the relationships in a lower dimension rather than profiling any group structure that may be present across the datasets. Rule based methods are also commonly used as profiling tools aimed at identifying groups over many databases. However, for reasonably sized datasets, these methods require summarising thousands of individual rules.

2.4.1 Multivariate regression and multivariate analysis of variance

Multivariate experimental design is the most common example of profiling in statistics. The model for both multivariate regression and analysis of variance (MANOVA) is stated in the form of a general linear model,

$$\underset{(N\times M)}{Y} = \underset{(N\times P)}{X}\underset{(P\times M)}{B} + \underset{(N\times M)}{U} \tag{2.12}$$

where B is a matrix of coefficients estimated by least squares and U is the corresponding error matrix. If X is a matrix of predictor variables then the model in (2.12) is identical to performing separate univariate multiple regressions for each response variable. However if X is substituted for an ANOVA design matrix, solving (2.12) is performing a multivariate hypothesis test. The major assumption of these models is that the error matrix U follows a multivariate normal distribution. Multivariate linear models of this type are designed to predict Y. They do not uncover common groupings unless they are known and coded into the design matrix.

2.4.2 Canonical correlation analysis (CCA)

Canonical Correlation Analysis (CCA) models common correlation structure between Y and X by summarising them in a lower dimensional space. These dimensions, $\underset{\sim}{u}_i$ and $\underset{\sim}{v}_i$ are weighted projections of Y and X:

$$\begin{aligned} \underset{\sim}{u}_i &= \underset{\sim}{a}_i^T Y \\ \underset{\sim}{v}_i &= \underset{\sim}{b}_i^T X \end{aligned}, \tag{2.13}$$

where $\underset{\sim}{a}_i$ and $\underset{\sim}{b}_i$ are called the canonical variates. The canonical variates are found in directions of decreasing maximum squared correlation r^2 between $\underset{\sim}{u}_i$ and $\underset{\sim}{v}_i$. The canonical variates are a mapping between Y and X. The maximum number of canonical variates that can be extracted is less than or equal to the smallest number of variables present in either *Y* or *X* (Seber 1984). These variates have mathematical relationships with the results of multivariate regression and MANOVA, but also with linear discriminate analysis and so may highlight group structure between the two datasets (Rencher 2002).

2.4.3 Procrustes analysis

Methods such as CCA require the specification of a response and predictor set, and as such only work for relating two datasets. Methods of combination like Generalized Procrustes Analysis and Individual Scaling Analysis (INDSCAL) (Carroll and Chang 1970) relate many datasets together in order to find a matching configuration. Over *M* datasets these methods minimise,

$$\min\left\{\sum_{m=1}^{M}\left\|\underset{\sim}{\omega}_m X_m - \bar{X}\right\|^2\right\} \tag{2.14}$$

where the matching configuration is $\bar{X}$ and ω_m is a weight vector that defines a rotation upon X_m such that it is closest to the mean configuration. This mean configuration highlights dominant structure over all data sources. However as the purpose of these methods is as a data reduction tool, it is difficult to identify the source of the observed groups.

2.4.4 Decision and association rules

Association rules are logical expressions found within a dataset that are characteristic of, or are related to an outcome. Used commonly in "*market basket analysis*" (Giudici 2003) the association rules are of the form,

$$d(x) \Rightarrow y \tag{2.15}$$

where *d(x)* is a decision rule on a variable *x* that logically implies *y*. The form of *d(x)* can be in the that of a single variable inequality or as a logical combination of inequalities using "and" ($\wedge$) and "or" ($\vee$) operations (Lent, Swami and Widom 1997). As this definition is quite flexible association rules can be easily defined for all data types.

For any realistic analysis thousands of individual association rules are identified. This means a way of filtering to find only interesting or significant association rules is needed. This brings forward the ideas of support and confidence for an association rule. For the simple rule $d(x) \Rightarrow y$ the support *S* and confidence *C* are defined as,

$$\begin{aligned} S &= P(A \wedge B) \\ C &= P(B|A) \end{aligned}, \tag{2.16}$$

where the support is the probability of both events *A* and *B* occurring together, and the confidence is the probability of event *B* given an event *A*. Identifying interesting rules is usually done by imposing a threshold on either the support or confidence on individual rules (Srikant and Agrawal 1997), or by constructing a weighted combination of rules based on how well they predict a response (Friedman and Popescu 2005).

A Classification and Regression Tree (CART) is a hierarchical combination of decision rules and is often referred to as a decision tree (Kass 1980, Breiman, et al. 1984). Extensions to CART for multivariate regression (MRT) (Yan Yu and Diane Lambert 1999, De'ath 2002, Larsen and Speckman 2004) allow for profiling a continuous response set, by a mixed predictor set. Auto-associative Multivariate Regression Trees (AAMRT) (Questier, et al. 2004) look for decision rules that partition a continuous profiling set into homogeneous clusters, allowing for MRTs to work both as a profiling and clustering tool. Furthermore these decision rules can then be used to cluster new observations. This also allows for validation regimes to be imposed over the model to test for validity and help estimate the number of groups present (Smyth, et al. 2005).

The flexibility of decision rule methods to handle most data types is their most powerful feature. When combined into a decision tree they provide an intuitive statistical framework to conduct profiling analysis. Furthermore ensembles of trees not only improve and stabilise the predictions of a single tree but also provide useful links with cluster ensemble methods (Section 2.4). Therefore tree-based methods are a natural choice for profiling analysis and are the focus of this thesis.

2.5 Tree-based Profiling

Tree based techniques present an ideal framework for profiling as they produce predictive decision rules on the predictor variables that identify group structure within the response. **The primary aim of this thesis is to extend tree based profiling to**

handle a mixed type response set. To achieve this goal several approaches are implemented in this thesis culminating in the development of MCTs.

Mixed type extensions to standard CART theory are proposed using the Gower dissimilarity and binary substitution. These methods transform the response dataset into a form suitable for use with the existing MRT or Db-MRT methods. Furthermore through the use of auto-association these techniques double as a clustering tool. The result of this is a single multivariate tree that can profile a simple mixed type response set.

Binary substitution represents a mixed type dataset set as a multivariate continuous dataset compatible with MRTs and therefore allows a simple mixed type extension to multivariate ensemble methods. In this thesis MRT theory with binary substitution is used to implement multivariate mixed type extensions to random forests and treeboost. Multivariate random forests and treeboost are predictive clustering or profiling techniques that can handle multivariate response and predictor sets and are resistant to a large number of variables within the predictor set.

Through the use of the proximity matrix arising from tree-based ensembles this thesis proposes MCTs, which is a new technique for either profiling or clustering. MCTs present methods to intelligently combine individual ensemble proximity matrices to allow for improved understanding of the structure within each response variable. By partitioning these combined ensemble proximity matrices MCTs provide a method to identify grouping profiles found over all ensembles. This allows for removal of unimportant response variables from the analysis and more importantly allow MCTs to identify subgroups of variables within the response set.

3. Methods

In this section is the core theory that forms the base of the MCT model. Firstly the CART methodology is described along with the relevant multivariate extensions MRT, AA-MRT and Db-MRT. This is followed by multivariate extensions to tree-based ensembles. From here, the link between tree-based ensembles and cluster ensembles is made by describing techniques to combine ensemble proximity matrices.

To assist the reader in interpretation of the models presented here an example implementation is provided on the benchmark iris dataset (Fisher 1936).

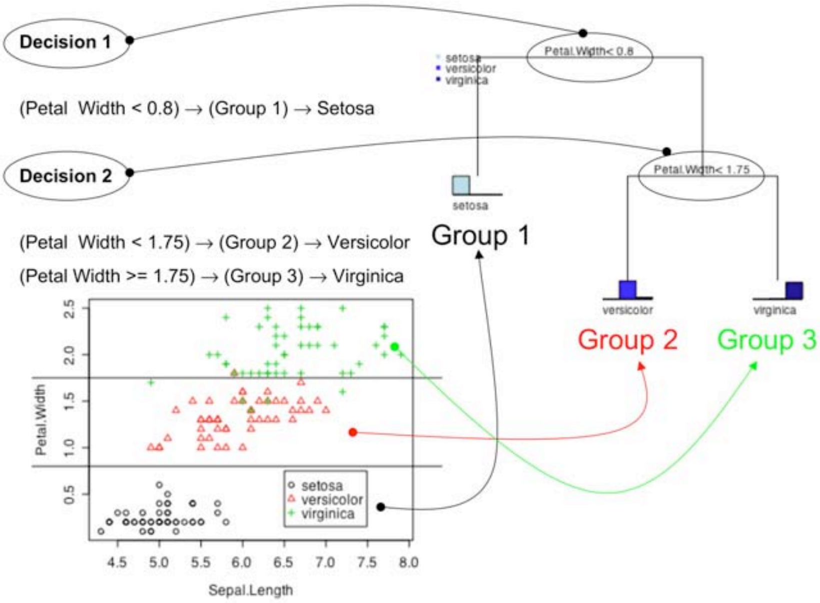
3.1 Tree-based Models

A decision tree is a hierarchy of decision rules that partition a response into separate groups (Figure 3). This hierarchy imposes interactions between decision rules using an "*and*" operation. Figure 3 presents an example decision tree to classify three varieties of iris flowers (Setosa, Versicolor and Virginica) based on their sepal and petal length and widths using the benchmark iris dataset. The first decision or split is found on the variable "*petal width*". The effect of this split is to partition the dataset into two mutually exclusive groups; the first containing iris flowers with a petal width less than 0.8 and the second with petal widths greater than 0.8. This decision has the effect of accurately defining the Setosa variety of iris flower in the left terminal node (3.1). From the scatter plot in Figure 3 it is obvious that based on petal width the Setosa variety of flowers is the most easily identified showing a considerable smaller petal width and is therefore the first split within the tree.

$$\begin{array}{l}
(1)\ \{(\text{Petal Width} < 0.8) \Rightarrow (\text{Group } 1) \Rightarrow \text{Setosa}\} \\
(2)\ \{(\text{Petal Width} >= 0.8) \wedge (\text{Petal Width} < 1.75) \Rightarrow (\text{Group } 2) \Rightarrow \text{Versicolor}\}\,. \quad (3.1) \\
(3)\ \{(\text{Petal Width} >= 0.8) \wedge (\text{Petal Width} >= 1.75) \Rightarrow (\text{Group } 3) \Rightarrow \text{Virginica}\}
\end{array}$$

From here the tree must further partition the dataset to identify the other two varieties of iris flowers. As the first split accurately determined Setosa in the left terminal node, the second split to determine Versicolor and Virginica must be on the right. By coincidence this split is also performed on petal width, however could have potentially been on any other predictor variable within the dataset. As the second split depends on the first, to determine the two other varieties of iris flowers the compound decision rules in (3.1) are necessary. By observation of the scatter plot in Figure 3 it can be seen that this decision is not as clear as the first. In fact, this split invokes a misclassification of 6 out of the 150 iris flowers.

Figure 3: Example decision tree classifying the species of flowers in the iris dataset.



In this example the tree built is a classification tree and the splits are performed on quantitative variables. However, the flexibility of tree based models comes from the easy definition of decision rules for either categorical or quantitative predictor variables, for either univariate classification or multivariate regression. There are two popular tree growing algorithms available, Classification and Regression Trees (CART) (Breiman, et al. 1984) and C4.5 (Quinlan 1993). Due to the easy accessibility of the "*rpart*" package (Therneau, Atkinson and Ripley 2005) in R (R Development Team 2005) to benchmark the CART algorithm, this framework is selected for use in this thesis.

3.2 Classification and Regression Trees (CART)

Building a CART model requires a search on two levels (Figure 4). Starting with all observations within the root terminal node, the first step is to find the next best split for each terminal node of the tree. This involves searching in every terminal node for the decision rule that minimises the impurity defined in (3.2). After all the next best splits have been found for each terminal node, a second search is performed over the terminal nodes of the tree. By minimising relative error (RE) of the tree defined in (3.3) the best node to split on is found. This split is then used to grow the tree. This process continues until no more splits can be found; this tree is called the maximal tree.

Figure 4: CART algorithm.

```
While tree size < maximum tree size do:
    1. Find the best split on each terminal node:
        a. For each predictor variable find the best split, d, by finding the
           minimum impurity R(d) (3.2).
        b. Over each predictor compare the best splits, and pick the variable
           with the smallest R(d). This split on this variable is the next best
           split for that terminal node.
    2. For each terminal node compute the RE(d) (3.3) using the next best split
       for that terminal node.
    3. Compare the RE(d) statistics over each terminal node and grow the tree on
       the minimum.
```

CART finds the next best split for each terminal node by ranking all possible decision rules within the predictor set. Each decision rule is assessed based on its impurity. The impurity of a decision is defined as the degree of heterogeneity of the response observations within each node resulting from the decision. Put more formally, the impurity *R(d)* of a decision *d* is defined as the weighted sum of the impurities of the left, $R\left(\underset{\sim}{y} \in \text{left}\right)$ and right $R\left(\underset{\sim}{y} \in \text{right}\right)$ nodes respectively,

$$R(d) = p_{\text{left}} R\left(\underset{\sim}{y} \in \text{left}\right) + p_{\text{right}} R\left(\underset{\sim}{y} \in \text{right}\right) \qquad (3.2)$$

where p_{left} and p_{right} are the probabilities of the left and right nodes respectively.

CART defines the idea of a relative error (RE) to assess the quality of the overall tree. The *RE(T*) of a tree *T,* is defined as the sum over the impurity of all terminal nodes,

$$RE(T) = \frac{\sum_{t_n \in T} R\left(\underset{\sim}{y} \in t_n\right)}{R\left(\underset{\sim}{y}\right)} \qquad (3.3)$$

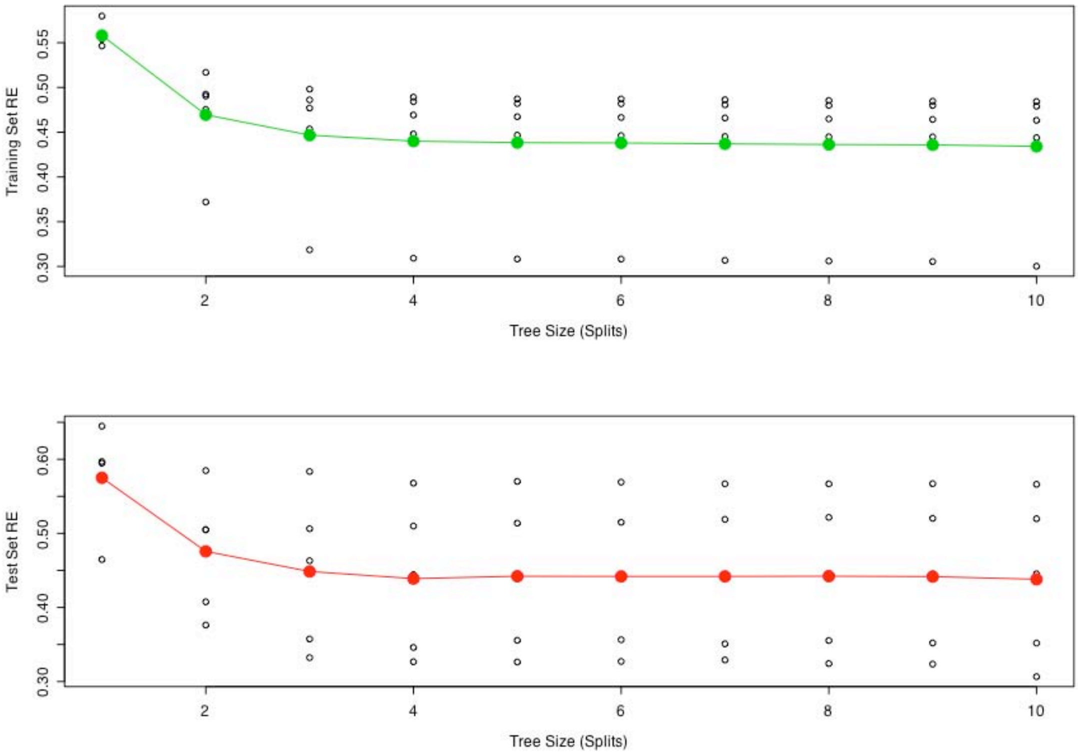
where t_n is a terminal node is tree *T* and $R\left(\underset{\sim}{y}\right)$ is the relative error of the non-partitioned response. As defined in (3.3) the *RE* is the percent error of a tree, and is a monotonically decreasing function.

3.2.1 Determining CART tree size

The algorithm in Figure 4 grows the maximal tree, and will result in a tree that overfits the data. Therefore a means of pruning the tree to a smaller size is needed. Tree size can be defined in two ways; either by specifying the minimum size of the terminal nodes or by specifying the maximum number of splits. The values of these parameters are most commonly estimated using V-fold cross validation. This involves the CART algorithm (Figure 4) to be run V times on subsets of the data. For each subset and for each tree size (1 split, 2 splits, … , etc.) the data not used to build the tree is predicted or classified. After the V iterations, the cross validated performances are presented on a graph called the relative error graph (Figure 5). The minimum of the cross validated RE graph is taken to be the optimal tree size for that dataset.

An example of a RE curve for a multivariate regression tree on the iris dataset is presented in Figure 5. This graph contains two plots corresponding the training set (top) and test (bottom) relative errors for tree sizes ranging from 1 to 10 splits. The reference line displays the mean performance for each tree size. For standard CART models 10-fold cross validation is usually implemented. The points surrounding the line are the individual performances for each tree over the course of 10-fold cross-validation. The spread of the points around the mean line at each tree size is related to how certain a tree of that size is. If the variance of the points is large then the structure of a tree grown to that size is uncertain. The task is now to estimate an appropriate tree size based on these graphs.

Figure 5: 10-fold cross-validated RE graph for the iris dataset.



Picking the optimal tree size can be done in many ways. The most common method is the "*1-SE*" rule. This rule defines the best tree to be the simplest tree with a RE within one standard error of the RE of the next tree. This implies that the RE of the next tree is essentially the same as the RE of the current tree, and then there is no improvement gained by growing the tree any further. The idea of the "*1-SE*" rule method is that the terminal nodes of the tree must be as stable as possible. As this thesis is focused finding stable groups within the terminal nodes of a tree, this method is employed to determine optimal tree size.

Minimal cost-complexity is another means to determine the size of CART models that is focused on optimal predictive performance (Breiman, et al. 1984). This rule estimates a penalising parameter that is a combination of the predictive performance of the tree and its complexity. The form of this penalty is,

$$RE_{\alpha}(T) = RE(T) + \alpha\left|\tilde{T}\right| \tag{3.4}$$

where $RE_{\alpha}(T)$ is a combination of its cost *RE(T)* and its complexity $\alpha\left|\tilde{T}\right|$ with $\left|\tilde{T}\right|$ being the number of terminal nodes in a tree and α is the estimated cost complexity parameter. The best tree is now picked at the minimum of (3.4) which is the minimum error tree given its size. Many other pruning algorithms exist for finding the optimal tree, such as Reduced Error Pruning (REP) and Pessimistic Error Pruning (PEP) (Quinlan 1987). For a comparison of the relative performances of these see (Esposito, Malerba and Semeraro 1997, Esposito, Malerba, Semeraro and Tamma 1999)

3.2.2 Finding the best split

Making a decision upon a variable requires an exhaustive search over all possible split points within each predictor variable. For continuous predictor variables sorting the response such that it is in the ascending order of the predictor variable, and then parsing it in this order will search all valid splits. In general for a continuous predictor variable there are *n* valid split points. For a categorical predictor variable with *k* levels a search over all $(2^{k-1}-1)$ possible splits is required. At each split point the impurity function, (3.2), must be evaluated and compared with the current best split. The specific impurity function used depends on the types of variables within the response set. Common impurity functions for classification and regression are now discussed.

3.2.3 Classification trees

The gini index for a split, *d*, is the sum of the gini indices for the left and right nodes resulting from the split,

$$\begin{aligned}R(d)=&\left(\frac{n_{Left}}{n}\right)\left(\sum_{\substack{k\in levels_y\\ y_i\in Left}}\left(p(y_i=k)(1-p(y_i=k))\right)\right)\\ &+\left(\frac{n_{Right}}{n}\right)\left(\sum_{\substack{k\in levels_y\\ y_i\in Right}}\left(p(y_i=k)(1-p(y_i=k))\right)\right)\end{aligned}, \qquad (3.5)$$

where $p(y_i=k)$ is the probability that an observation y_i is of class *k* within the left or right nodes, $levels_y$ is a list of all the groups within *y*, n_{Left} and n_{Right} are the number of observations in the left or right nodes respectively and *n* is the total number of observations . It is possible to simplify (3.5) further and get a more interpretable from of the gini index:

$$R(d)=\left(\frac{n_{left}}{n}\right)\left(1-\sum_{\substack{k\in levels_y\\ y_i\in Left}}p(y_i=k)^2\right)+\left(\frac{n_{Right}}{n}\right)\left(1-\sum_{\substack{k\in levels_y\\ y_i\in Right}}p(y_i=k)^2\right). \qquad (3.6)$$

As indicated in (3.6), minimising the gini index requires either $p(y_i=k)$ to be close to 1. In other words minimising the gini index will find terminal node class profiles with probabilities of each class being close to 1. Taken to its extreme this usually results in terminal nodes containing y_i's of the same class. After the tree is grown, classification trees use the highest probability class within a terminal node to classify the observations within that node.

3.2.4 Univariate regression trees

Regression trees minimise the squared error between all observations and their mean within a terminal node,

$$R(d) = \left(\frac{n_{Left}}{n}\right) \sum_{y_i \in Left} \left(y_i - \overline{y}_{Left}\right)^2 + \left(\frac{n_{Right}}{n}\right) \sum_{y_i \in Right} \left(y_i - \overline{y}_{Right}\right)^2 \qquad (3.7)$$

where $\overline{y}_{Left}$ and $\overline{y}_{Right}$ are the means of the observations in the left and right partitions respectively. In implementing (3.7), the squared error must be computed for all possible splits in each predictor variable. This is time consuming and not necessary as (Therneau, et al. 2005) show that maximising the within sums of squares over the entire split (both left and right terminal nodes) produces the same split and is considerably faster ,

$$\begin{aligned} &\min\left[\left(\frac{n_{Left}}{n}\right) \sum_{y_i \in Left} \left(y_i - \overline{y}_{Left}\right)^2 + \left(\frac{n_{Right}}{n}\right) \sum_{y_i \in Right} \left(y_i - \overline{y}_{Right}\right)^2\right] \propto \\ &\max\left[\left(\frac{n_{Left}}{n}\right)\left(\overline{y}_{Left} - \overline{y}_t\right)^2 + \left(\frac{n_{Right}}{n}\right)\left(\overline{y}_{Right} - \overline{y}_t\right)^2\right] \end{aligned} \quad , \qquad (3.8)$$

where $\overline{y}_{left}$ and $\overline{y}_{right}$ is the mean of the left and right terminal nodes of the split respectively and $\overline{y}_t$ is the mean of all observations in the parent node.

From (3.8) it can be seen that a split in regression trees can be viewed as either finding the maximum difference between the terminal node means and the total means; or, by (3.7), the same split is found by minimising the variance within each terminal node. The overall prediction made by a regression tree is the mean of each terminal node.

3.2.5 Multivariate regression trees (MRT)

Multivariate regression splitting (MRT) is simply the multivariate extension to (3.7) (Segal 1992, Yan Yu, et al. 1999, De'ath 2002, Larsen, et al. 2004),

$$R(d) = \left(\frac{n_{Left}}{n} \right) \sum_{y_i \in Left} \sum_{m=1}^{M} \left(y_{im} - \bar{y}_{Left,m} \right)^2 + \left(\frac{n_{Right}}{n} \right) \sum_{y_i \in Right} \sum_{m=1}^{M} \left(y_{im} - \bar{y}_{Right,m} \right)^2 , \quad (3.9)$$

where $\bar{y}_{Left,m}$ and $\bar{y}_{Right,m}$ are the means of the m^{th} response variable in the left or right nodes respectively. Minimising (3.9) is analogous to maximising the Mahalanobis distance with a covariance matrix equal to the identity (Segal 1992). Furthermore if the response matrix *Y* is a binary indicator matrix for a categorical variable as in (2.5), it can be shown that minimising (3.9) is the same as minimising the gini index (3.6) (Breiman, et al. 1984). This improves our understanding of the gini index: in that for classification splitting it directs the algorithm towards finding the split that minimises the variance of the probabilities for each level within a node (Hastie, Tibshirani and Friedman 2001).

MRTs offer a method for *a continuous profiling set and a mixed predictor set* because the tree is identifying groups (terminal nodes) within a response matrix *Y* that are defined by the predictor set *X*. MRTs identify stable and reproducible clusters, as the terminal nodes must be predictive of the response. Furthermore, the elbow of the relative error curve (Figure 5) which is used to estimate tree size also gives an estimate of the number of groups in the profiling solution (Smyth, et al. 2005). This is a validation regime over the profiling technique because the *RE* curve provides a cross validated procedure for estimating the number of groups based on predictive performance. This links in with the concepts of cross-validation (Dudoit, et al. 2003) and predictive validation (Dudoit, et al. 2002, Tibshirani, et al. 2005) for cluster validation.

3.2.6 CART on a distance matrix (Db-MRT)

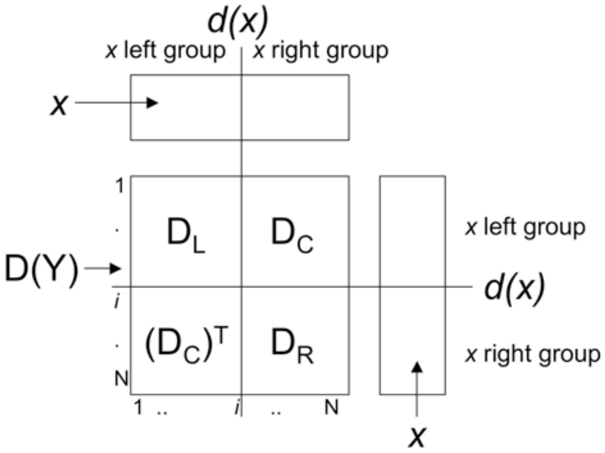
A distance matrix is a specific data type that summarises relationships between observations within a dataset. It is a square symmetric matrix of the form,

$$D(Y)=\begin{bmatrix} 0 & d\left(\underset{\sim}{y}_1,\underset{\sim}{y}_2\right) & \cdots & d\left(\underset{\sim}{y}_1,\underset{\sim}{y}_N\right) \\ d\left(\underset{\sim}{y}_2,\underset{\sim}{y}_1\right) & 0 & & \\ \vdots & & \ddots & \\ d\left(\underset{\sim}{y}_N,\underset{\sim}{y}_1\right) & \cdots & & 0 \end{bmatrix} \qquad (3.10)$$

where *D(Y)* is a distance matrix of size *n* by *n*, where *n* is the number of observations, $d\left(\underset{\sim}{y}_i,\underset{\sim}{y}_j\right)$ the distance between observations *i* and *j* in the response dataset *Y*. Forming splits on the distance matrix representation of the response set has been suggested as a flexible multivariate extension to CART (De'ath 2002).

As a distance matrix contains the observations on both the rows and columns, a split must also act on both. Figure 6 shows a distance matrix *D(Y)* is partitioned by a decision *d(x)* on predictor variable *x*. This results in four sub-matrices corresponding to the left group (D_L), right group (D_R) and the between group distance matrices which for ease of understanding in this thesis are called the covariance groups, (D_C) and $(D_C)^T$. The goal of a partition is to minimise the distances between the observations within both the left and right groups.

Figure 6: Example distance matrix partition.



Distance Based MRT (Db-MRT) (De'ath 2002) defines the node impurity as the sums of the squared distances within the left and right groups,

$$R(d)=\frac{1}{n_L^2}\left(\sum_{i\in D_L}\sum_{j\in D_L}d\left(\underset{\sim}{y}_i,\underset{\sim}{y}_j\right)^2\right)+\frac{1}{n_R^2}\left(\sum_{i\in D_R}\sum_{j\in D_R}d\left(\underset{\sim}{y}_i,\underset{\sim}{y}_j\right)^2\right) \tag{3.11}$$

which is exactly equivalent to standard MRT (3.9) if the distance metric between two observations is squared Euclidean, however takes no account of the distance between the observations of the two groups found in D_C. If the distance in (3.11) is Euclidean, the squared distance between the left and right group centroids can be defined by the Gower distance (Gower and Hand 1996),

$$d\left(D_L,D_R\right)=\bar{D}_L+\bar{D}_R-2\bar{D}_C \tag{3.12}$$

where $\bar{D}_L$, $\bar{D}_R$ and $\bar{D}_C$ are the centroids for each sub-matrix and are defined to be,

$$\bar{D}_g=\frac{1}{n_g^2}\sum_{i\in g}\sum_{j\in g}d\left(\underset{\sim}{y}_i,\underset{\sim}{y}_j\right), \tag{3.13}$$

where g denotes either the left, right or covariance sub-matrices. Either maximising (3.12) or minimising (3.11) is simply stating that the distances between objects within a cluster must be small. However it does not necessary follow that minimising (3.11) will maximise (3.12), due to the inclusion of the covariance group centroid in (3.12).

Tree based-distance splitting has the ability to profile group structure within a multivariate response. Furthermore it allows for profiling over a mixed type through the use of the Gower dissimilarity to construct the base matrix to be partitioned. However as a distance matrix is an abstraction upon the data, assessing predictive performance is difficult. This difficulty extends to problems in determining which response variables are expressed within a terminal node.

3.2.7 Auto associative multivariate regression trees (AA-MRT)

Trees can be considered a search for homogeneity within the response. Multivariate regression and Euclidean Db-MRT have strong relationships with K-Means and Wards method for agglomeration, as they all define a group by minimum within-group sums of squares. However unlike the other techniques MRTs use a predictor set to identify the groups.

Auto-Associative MRTs (AA-MRT) (Questier, et al. 2004, Smyth, et al. 2005) extend MRTs to be able to cluster a single dataset. The idea simply mirrors the response set within the predictor set of the MRT model. For example, to cluster a dataset *Y*, AA-MRT will grow a tree with *Y* as the response and predictor dataset. AA-MRT is a form of constrained K-means as the groups are found to reduce the within-sums-of-squares but also must be defined by the decision rules of the tree.

3.3 Ensembles of Trees

When modelling large datasets it is necessary to pick a model that can accurately assess the predictive performance of all predictor variables. Usually this is done as a two step procedure by first using a data reduction method, such as a partial least squares (De Jong 1993) or a genetic algorithm (Mitchell 1998). The output is then passed into a more powerful method for example see Hancock et al. (2005). Another alternative is to use a penalising method such as ridge regression (Hoerl and Kennard 1970) or penalised discriminate analysis (Hastie, Buja and Tibshirani 1995). Penalising methods impose strong conditions on estimation to reduce the risk of overfitting. However it is rapidly becoming clear that a single model is insufficient for analysing large datasets. More commonly, collections of models are combined into an ensemble to create an overall large model. By doing this, statisticians are treating a single model more as a variable within a larger modelling scheme.

A weak learner (Schapire 1990) is a model that is guaranteed to perform better than a coin flip. CART falls into this category because the *RE* must be a decreasing function. Therefore the worst possible performance of CART must still outperform the mean variation on the training set. As encouraging as this is, it is well known that the predictive or classification performance of CART is average to poor (Hastie, et al. 2001). However, weak learners like CART are ideal for ensemble methods, as it is known that any individual model produced must do better than random chance.

Ensemble methods combine the results of many weak learners to improve their overall predictive performance (Breiman 2001). Commonly behind these techniques is the idea of bootstrapping to improve the predictive performance of the model (Efron 1979). By taking random samples of the training set many different models can be built, each selecting different variables and displaying different characteristics. These models are then combined together using a simple linear combination. There are many different types of ensembles; the differences between them lie in how the linear combination of models is constructed. Two common means of building ensembles are *bagging and boosting*.

3.3.1 Bagging, random forests and multivariate random forests

Bagging (**B**ootstrapped **Ag**gregation) (Breiman 1996a), is the simplest means of building an ensemble. Bagging averages the results over many bootstrapped models. For discriminate analysis, the classification of a single observation is the majority vote over all bootstrapped models and for regression it is simply the mean prediction. Bagging usually performs better than a single model (Breiman 1996a), and also improves the accuracy of the variable importance statistics (Breiman 1996a, Dietterich 2000b).

A common extension of bagging is implemented using a CART model and in this form, it is called Random Forests (RF) (Breiman 2001). The difference is that the random forest bootstrap is performed over the variables and the observations simultaneously (Figure 7). This ensures that each tree has the best chance of being different. The more different the trees, the better the bagging model will perform (Breiman 1996a, Dietterich 2000b, Friedman and Popescu 2003).

How different each tree is from the others is called the diversity of a forest. The diversity of the forest has a direct relationship with an upper bound generalization error, *PE* and is given by,

$$PE \le \frac{\bar{\rho}\left(1 - s^2\right)}{s^2} \tag{3.14}$$

where $\bar{\rho}$ is the mean correlation between trees in the forest and *s* is the strength of the random forest classifier defined as,

$$s = E_{X,y}\left(\text{margin}\left(\text{X,y}\right)\right), \tag{3.15}$$

where *X* are the predictors and *y* is the response. The margin is the estimate of by how much the predictions of the forest exceed random chance (Breiman 2001).

By (3.14), it is shown that, as the correlation between the trees in the forest increases, the upper bound on the error of the forest also increases. As diversity is measured by the mean correlation, $\bar{\rho}$, between the trees of the forest, the higher the diversity the lower the mean correlation and consequently the lower the error of the forest. Counteracting this in (3.14) is the relationship between $\bar{\rho}$ and *s*, where *s* acts as a limiting factor on *PE* by resisting the increase caused by increasing $\bar{\rho}$. If $\bar{\rho}$ of the trees within a forest increases, each tree is identifying similar structure within the response variable. The parameter *s* is then to assess how correct that structure is and to penalize the model accordingly. The action of bootstrapping on the construction of the trees is intended to minimise $\bar{\rho}$, which then allows (3.14) to be dominated by *s*. It is expected that (3.14) also holds as a loose upper bound over any ensemble learner (Breiman 2001).

The algorithm for random forests is simple and is given in Figure 7. The user specifies the number of trees in the forest, the number of observations and variables to be randomly selected to build each tree and the tree building parameters. One important addition to the random forest model is the use of “out-of-bag” samples to assess model convergence. Out-of-bag (OOB) samples are those observations that are in the training set, but not in the bootstrapped sample which was used to build the tree. Using the OOB sample to estimate the error rate of the forest will provide a more realistic estimation of this parameter.

Figure 7: Random forests algorithm.

```
1. While the number of trees < maximum number of trees do:
        a. Take a random sample of observations.
        b. Take a random sample of variables.
        c. Grow a maximal tree.
        d. Predict the left out observations.
        e. Update OOB, test and training set errors.
```

In this thesis random forests are extended to multivariate regression by implementing the algorithm in Figure 7 with multivariate regression trees (Section 3.2.5). Furthermore, by binary substituting the categorical variables as described in Section 2.2, multivariate random forests are extended to handle a mixed type response set.

3.3.2 Auto associative random forests and the random forest proximity matrix (RFP)

The natural tendency of tree-based methods is to find predictable groups within large datasets. Random forests can be considered as an search to find every possible tree that can be formed. By combining the two ideas, random forests can be seen as bagging a clustering algorithm, with each tree finding a slightly different grouping structure within the data. From this it is possible to form a proximity matrix on the observations of the data, mapping their grouping tendency (Breiman 2001). This matrix contains a similarity between the response variable observations as seen by the profiling set. This matrix is:

$$C = \begin{bmatrix} N_B & c_{12} & \cdots & c_{1n} \\ c_{21} & N_B & & \\ \vdots & & \ddots & \\ c_{n1} & & & N_B \end{bmatrix} \tag{3.16}$$

where c_{ij} is the number of times the cases *i* and *j* have been placed into the same terminal node in every tree within the forest, and N_B is the number of trees within the forest.

Observation of the grouping structure within *C* is best viewed with an metric multidimensional scaling plot (MDS) (Gower, et al. 1996, Breiman 2001). An example of *C* can be found in Figure 8. This proximity matrix is a very powerful profiling feature of random forests as it allows a visual representation of how the predictor set views the groups in the profiling set. Furthermore, it can be constructed over a mixed type predictor set. Because of these features the RFP is a cornerstone idea behind the MCT method developed in this thesis.

Unsupervised random forests (Shi and Horvath 2003) provide a means of generating these proximities over a single dataset without a response dataset, such that the proximity matrix can be used for clustering. To do this a simulated response is constructed. This response is a categorical variable where all observations in the original data are labelled as ‘1’. The original data set is then inflated with new observations created by taking random samples from the marginal distributions of the original variables. This sampled data is then labelled as ‘2’ in the response. Random forests are the run to classify the response group. As there should be no difference between the original and sampled data then a decision to partition the response into subgroups may indicate prominent group structure within the predictor variables. Therefore the trees grown within unsupervised random forest will reflect the grouping structure variables within the predictor set. The proximities from this process can then be used in other clustering methods. This idea of using a cluster models to vote on interobject distance has also been used by (Dudoit, et al. 2003) and consensus clustering (Monti, et al. 2003).

Unsupervised RF however has the problem that predicting a simulated response makes little sense. As in thesis multivariate random forests have been developed (Section 3.3.1) it is possible by the idea of AA-MRT (Section 3.2.7) to also implement Auto-Associative Random Forests (AA-RF). This extension allows for the construction an RFP over a meaningful response. The difference between the two approaches is best described in an example using Fisher’s benchmark iris dataset (Figure 8) (Fisher 1936). More so, by binary substituting the response set of AA-RF it is possible to extend it to handle mixed data types.

AA-RF has the advantage of observing the group structure within the response and the predictor variables, which results in a significantly clearer proximity image with the three groups being obvious along the diagonal (Figure 8). This is translated into a clear MDS representation for AA-RF (Figure 8) where the groups ‘vericolor’ (2) and ‘virginica’ (3) are less overlapping than in unsupervised random forests. Furthermore, an analysis of the predictive performance of the AA-RF can be observed (Figure 9) to assess the accuracy of the groups found in the MDS plots.

Figure 8: A comparison between the unsupervised random forest and AA-RF proximities. The proximity matrices have been re-ordered by the known iris groups: (1) Setosa, (2) Vericolor, (3) Virginica. Yellow represents a high count between the observations, red a low count.

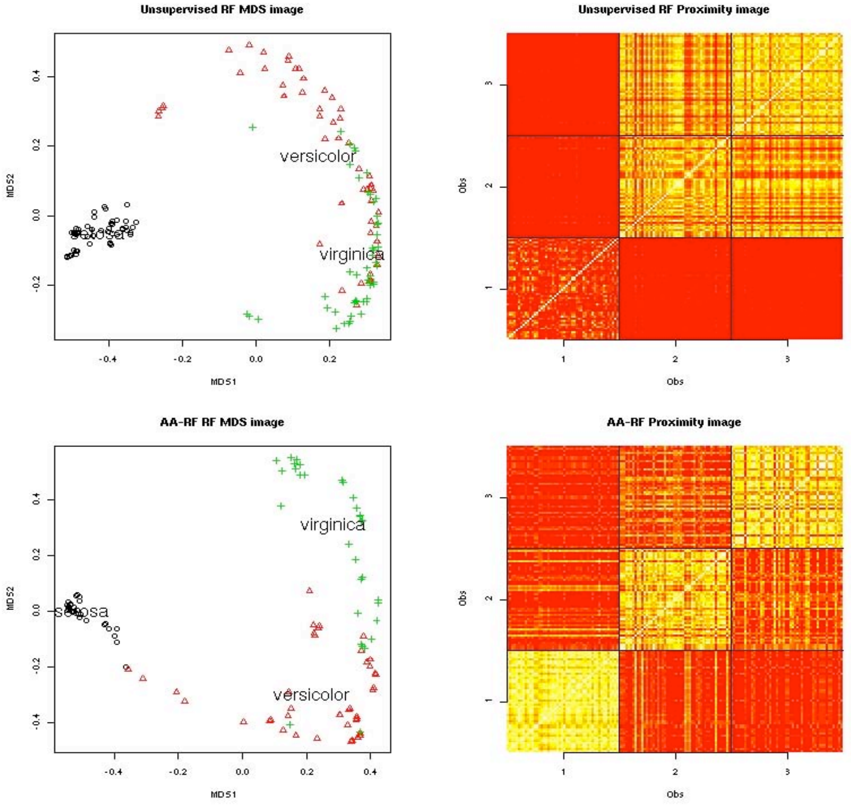
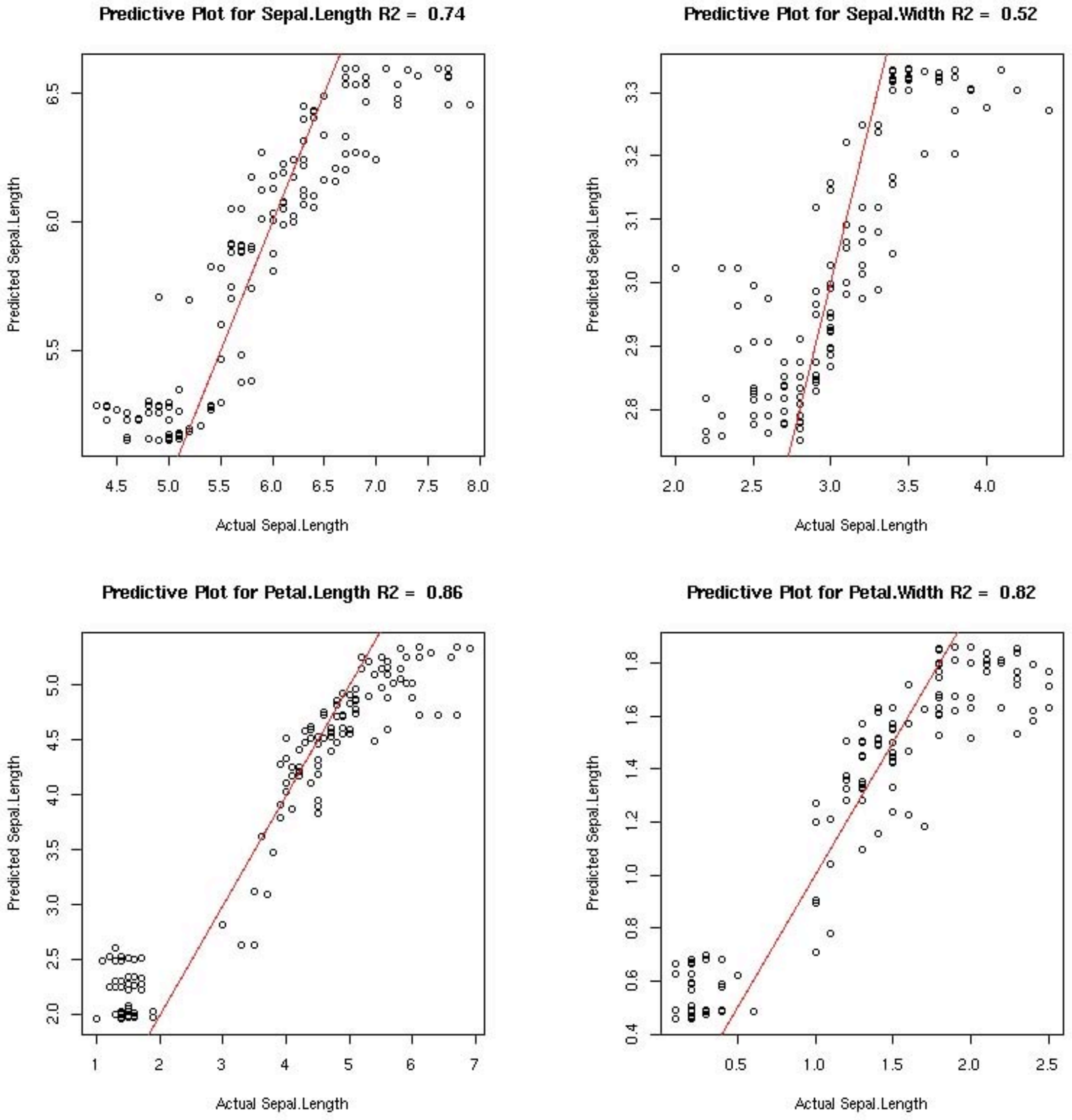


Figure 9: AA-RF predictive performance with predictions on the y-axis and actual variables on the x-axis and a reference line running through y = x. The multivariate R^2 = 0.97 and the individual variable R^2s are printed in the titles of each plot.



3.3.3 Boosting, treeboost and multivariate treeboost

Boosting is a stage-wise addition of models into a linear combination. Unlike random forests where each tree is built independently from the other, boosting conditions each new tree on the performance of past trees. A boosting model $f_m(x)$ forms iteratively as a recurrence relation, such that after *m* iterations,

$$f_m(x) = f_{m-1}(x) + \beta_m h\left(x;\underset{\sim}{a}_m\right) \tag{3.17}$$

where β_m is the coefficient of the new model $h(x;\underset{\sim}{a}_m)$ with parameters $\underset{\sim}{a}_m$, to be added to the previous boosting model $f_{m-1}(x)$. The complexity of boosting lies in how the model weights, β_m, are estimated.

Adaboost (Freund and Shapire 1997) was the first boosting algorithm developed and it displayed improved results for binary classification. Stochastic Gradient Boosting (MART) (Friedman 1999, 2001) is a faster, more accurate method for constructing a boosted model for classification or regression problems. MART conditions each new model to lie along the path of steepest decent of the loss function *L*. MART minimises,

$$\underset{\underset{\sim}{a}_m}{\arg\min} \left\{ L\left(y, f_{m-1} + \upsilon h\left(x;\underset{\sim}{a}_m\right)\right) \right\}, \tag{3.18}$$

where $h\left(x;\underset{\sim}{a}_m\right)$ is the next tree with parameters $\underset{\sim}{a}_m$ in the boosted set found and ν is the shrinkage factor. MART models are also easily defined for both regression and classification problems.

For regression, MART estimates the new model $h\left(x;\underset{\sim}{a}_m\right)$ to predict the residuals of the previous boosted model:

$$\underset{\underset{\sim}{a}_m}{\arg\min}\left\{L\left(y-\hat{y}_{m-1},h\left(x;\underset{\sim}{a}_m\right)\right)\right\}, \tag{3.19}$$

and in this way each model is forced to lie in the direction of steepest decent of the loss function. For classification, the new model is found to predict the residuals within the probability domain of the response,

$$\underset{\underset{\sim}{a}_m}{\arg\min}\left\{L\left(I\left(y_i=k\right)-\left(p_k\left(\underset{\sim}{x}_i\right)\right)_{m-1},h\left(x;\underset{\sim}{a}_m\right)\right)\right\} \tag{3.20}$$

where $I(y_i = k)$ is '1' if an observation y_i is of class k, '0' otherwise, and $\left(p_k\left(\underset{\sim}{x}_i\right)\right)_{m-1}$ is the probability that y_i belongs to class k given the current boosted model. Interestingly (3.20) is analogous to (3.19) where the sums of squares are computed on the probabilities that each observation belongs to each class.

MART finds $h\left(x;\underset{\sim}{a}_m\right)$ on a bootstrapped sample of the observations within the training set. The shrinkage parameter ν arises from the regularisation of boosting model such that it is resistant to overfitting. Unlike random forests, the trees grown in a boosted set are small, commonly stumps (one split). The inputs into a MART algorithm (Figure 10) are the tree size, the number of trees, the shrinkage parameter and the tree building parameters.

Figure 10: Treeboost algorithm.

```
1. While number of trees < max number of trees do:
     a. Take a random sample of observations and variables.
     b. Compute the residuals of the current model, $f_m$.
     c. Fit a tree predicting these residual to get $h(x;a_{m+1})$.
     d. Update the boosting model by $f_{m+1} = f_m + \nu h(x;a_{m+1})$.
```

Treeboost can be extended to multivariate regression by implementing the algorithm in Figure 10 with multivariate regression trees (Section 3.2.5) (Sain and Carmack

2002). Furthermore, by binary substituting the categorical variables as described in Section 2.2, multivariate treeboost is extended to handle mixed types within the response set.

3.3.4 Auto associative treeboost and the treeboost proximity matrix

In the same way as AA-RF (Section 3.3.3) extends AA-MRT (Section 3.2.7) in this thesis multivariate treeboost is extended to Auto-Associative Treeboost (AA-Treeboost). AA-Treeboost models are used for finding groups within a single dataset through the construction of a treeboost proximity matrix. The treeboost ensemble proximity matrix is constructed in an identical way to the random forest proximity matrix (Section 3.3.2), but over the trees of a boosted ensemble. This allows for a comparison in the ensemble proximities performances of AA-Treeboost and AA-RF. More so, by binary substituting the response set of AA-Treeboost it is possible to extend it to handle mixed data types.

In the iris example (Figure 11), the boosted proximity matrix is comparable to the unsupervised random forests image and the MDS plot shows high levels of overlap between vericolor (2) and virginica (3). Despite this it can be seen from the individual predictive plots that treeboost (Figure 12) outperforms random forests (Figure 9) on all variables but Petal length.

However, because of the linear form of the treeboost model, the structure within boosted proximity matrix may be undefined, as the construction assumes an equal contribution of all trees within the ensemble. As boosting applies a weight to each tree this assumption is not valid. Furthermore, as a multivariate response is more complex than a univariate response more care is needed in specifying the shrinkage parameter to avoid overfitting. It is expected that boosting will out perform random forests (Breiman 2001) but understanding the resulting model is considerably more difficult.

Figure 11: AA-Treeboost proximity results. The proximity matrices have been re-ordered by the known iris groups: (1) Setosa, (2) Vericolor, (3) Virginica. Yellow represents a high count between the observations, red a low count.

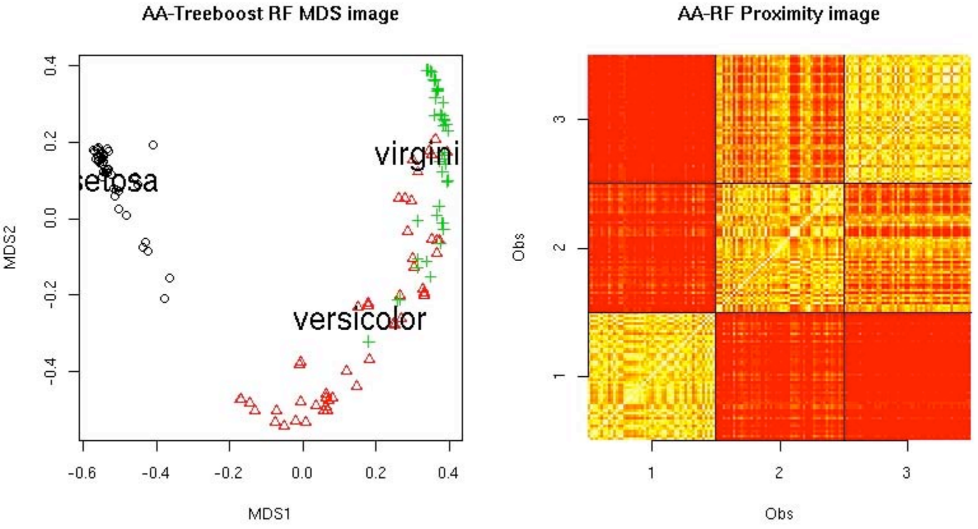
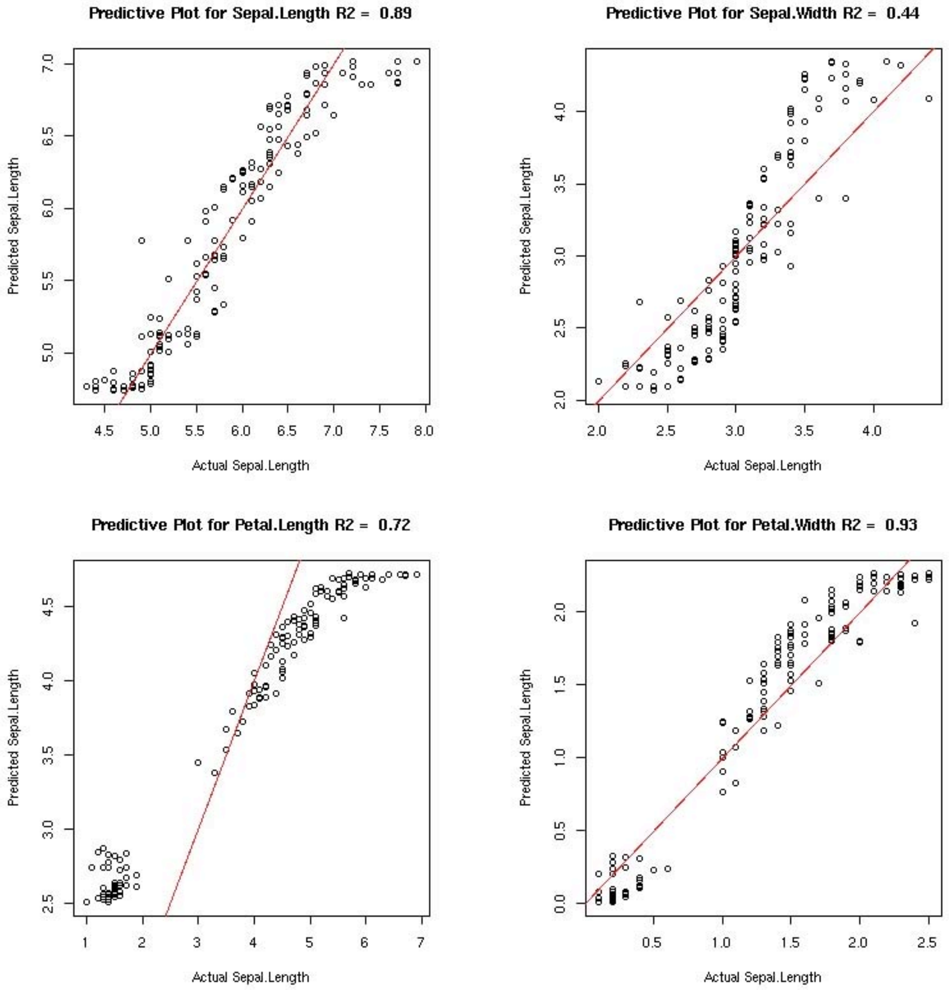


Figure 12: AA-Treeboost predictive performance on the iris dataset with predictions on the y-axis and actual variables on the x-axis and a reference line running through y = x. The multivariate R^2 is 0.996, and the individual variable R^2s are printed in the titles of each plot.



3.3.5 Interpreting ensembles

Ensembles work because they make use of a majority vote on the prediction of an observation (Dietterich 2000a) . In this interpretation, random forests assign each tree a unit vote on the prediction, and boosting defines the vote as a weight where, the higher the weight, the more influence that tree has on the model. This understanding of ensembles relies on the flexibility of “weak learners”. If you are assured that a model will always predict the training set to some degree, then it is possible to assign a weight or vote to that model based on your confidence.

As ensemble models can involve a large number of individual models, simply understanding the weights for each model is difficult. A lot has been made of this paradox of ensembles that as you increase the number of models within an ensemble to well in excess of the number of observations, the testing set error does not increase. By the sheer size of the ensemble methods it is expected that overfitting would result. As noted by (Schapire, Freund, Bartlett and Lee 1998), this flies in the face of model parsimony theories.

Because of this, much of the work surrounding the understanding of ensembles has centered on this conundrum by trying to find bounds on the generalisation error (Freund, et al. 1997, Schapire, et al. 1998, Breiman 2001, Schapire 2002). In doing this, an ensemble is treated as a method for optimising the performance of a particular model. This theoretical work goes to the heart of how these models optimise the solution, but pays little attention to how the final result may be understood.

Another interpretation of an ensemble is as a means of simulating the distribution of models, where different combination methods are simply different ways of simulating a likelihood distribution (Friedman, Hastie and Tibshirani 2000). This idea links ensembles with methods like Bayesian Model Averaging (BMA) (Raftery, Madigan and Hoeting 1997). BMA imposes a weight upon the models of an ensemble depending on where they occur within the known model distribution. This can be considered as a post-processing step over the ensemble in an effort to improve their performance (Friedman, et al. 2003).

3.3.6 Multidimensional Scaling (MDS) representation of ensemble proximity matrices.

Classical multidimensional scaling (MDS) (Torgerson 1958) is used in this thesis to view the group structure within the ensemble proximity matrices. MDS seeks a lower dimensional representation of a dissimilarity matrix whilst preserving the pair wise distances as much as possible. Each new dimension, z_d is found such that it minimises the stress function,

$$Stress\left(z_1, z_2, \cdots, z_D\right) = \left[\sum_{i \neq j}\left(d_{ij} - \left\|z_i - z_j\right\|\right)\right]^{1/2} \tag{3.21}$$

where d_{ij} are the ensemble proximity dissimilarities found by,

$$d_{ij} = N_B - s_{ij} \tag{3.22}$$

where N_B is the number of trees within the forest. In this thesis, as MDS is used only as a visualisation tool the representation is only to 2 dimensions.

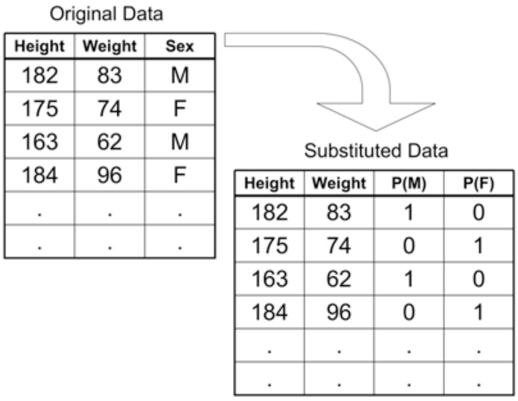
3.4 Issues With CART For Mixed Type Responses

The major goal of this thesis is to extend CART to handle mixed types within the profiling set. To do this two benchmark approaches are described. **Substitution of categorical response variables as a binary indicator matrix is the first benchmark mixed type extension to CART**. In effect this is making each group within a categorical variable a response variable (Figure 13). From here the substituted response set is treated as a multivariate regression problem, and the tree is grown with MRT. This can be done because it is known that minimising the gini index in (3.6) on a categorical variable is the same as minimising the multivariate sums of squares (3.9) on the indicator matrix representation of the response (Breiman, et al. 1984)(p. 124-125). Therefore by using MRT to find a split on a substituted matrix, the impurity minimisation is the sum of the multivariate sums of squares and the gini indices,

$$\begin{aligned} R(d) = & \left(\frac{n_{Left}}{n}\right)\left(\sum_{j \in Continuous} SS\left(\underset{\sim}{y}_{i \in Left,j}\right) + \sum_{j \in Categorical} GINI\left(\underset{\sim}{y}_{i \in Left,j}\right)\right) \\ & + \left(\frac{n_{Right}}{n}\right)\left(\sum_{j \in Continuous} SS\left(\underset{\sim}{y}_{i \in Right,j}\right) + \sum_{j \in Categorical} GINI\left(\underset{\sim}{y}_{i \in Right,j}\right)\right) \end{aligned} . \qquad (3.23)$$

A problem with this method arises when the categorical responses have a large numbers of levels. This results in inflating the number of variables in response set by the total number of the levels in all categorical variables. Furthermore if some levels of the categorical variables are not well represented within the observations of the response set, a sparse matrix will result where accurate prediction is difficult. Also care should be taken to standardize the continuous response variables before analysis to reduce any bias towards them.

Figure 13: Binary substitution of an example dataset where the nominal type response variable ‘Sex’ becomes two binary variables *P(M)* and *P(F)* corresponding to the probability of a person being male and female respectively.



Original Data

Height	Weight	Sex
182	83	M
175	74	F
163	62	M
184	96	F
.	.	.
.	.	.

Substituted Data

Height	Weight	P(M)	P(F)
182	83	1	0
175	74	0	1
163	62	1	0
184	96	0	1
.	.	.	.
.	.	.	.

The second benchmark method uses Db-MRT with the Gower dissimilarity (2.2) to grow a tree over a mixed type. It is expected that the Gower dissimilarity and binary substitution will produce similar results as both assume a Euclidean relationship between categorical and continuous variables and homogenous levels within a categorical variable.

With profiling analysis we are not so much interested in predicting the responses as accurately as possible, but finding stable groups within them. It is natural to assume that in a multivariate response set, different variables will show different grouping structure (Figure 2), where the final profiling analysis will be a compromise over all structures. There is considerable interest when performing profiling analysis to be able to assess the influence of each variable on the final group structure. Unfortunately with binary substitution and Db-MRT, this can be a difficult process because it is a simultaneous model over all variables that leaves little room for the analysis of individual effects.

The random forest proximity matrix (RFP) (3.16) is a summary of the groups found over many trees. This is a similarity matrix as in consensus clustering (Ben-Hur, et al. 2002, Monti, et al. 2003) and can be constructed for regression or classification analyses. This thesis exploits this flexibility of the RFP for mixed type analysis.

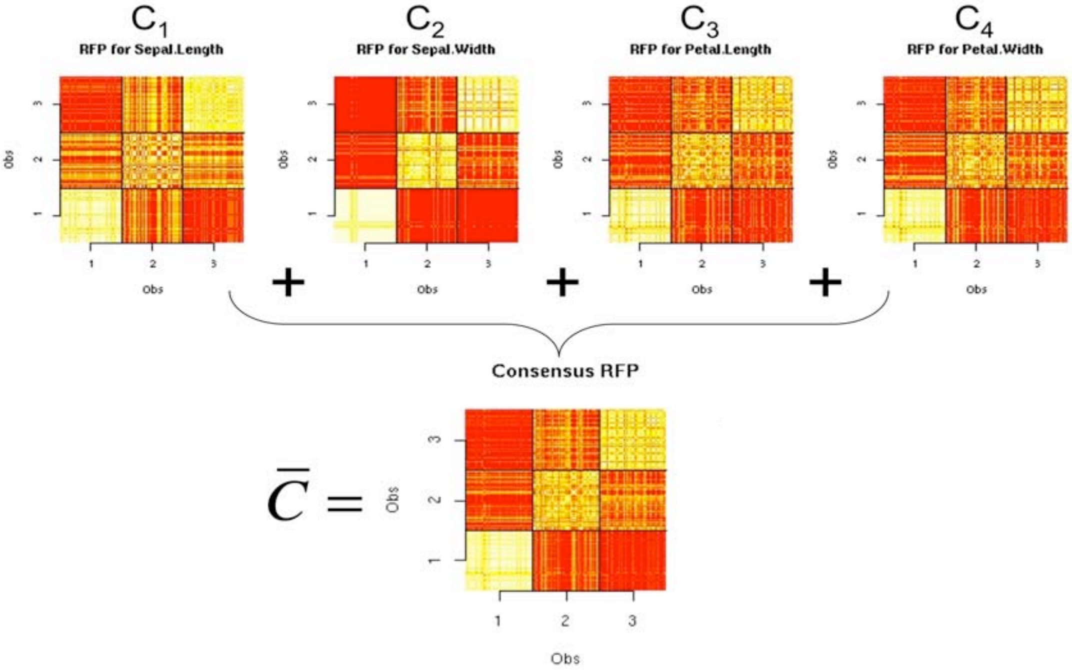
An RFP is a representation of the group structure within a variable of any type. This thesis proposes three methods to intelligently combine individual RFPs into one overall consensus RFP. This overall consensus RFP will provide a summary of the group structure over many variable of mixed type. By combining RFPs in this way is building an ensemble of RFPs is analogous in concept to cluster ensembles.

This overall consensus matrix is ideal for partitioning with a tree, because it is formed by combining the results from many random forests, therefore is a summary of all tree-based grouping structures over all variables. Growing a tree from a consensus matrix is expected to highlight the common groups found over all trees.

3.5 Combining Proximity Matrices

When considering multiple responses MCTs grow a random forest for each response variable independently to get a set of M RFPs, $\{C_1,\cdots,C_m,\cdots,C_M\}$. The group structure of each is then summarised and combined into the one overall consensus matrix, $\bar{C}$. If only one dataset is present then treating each variable as a response, and predicting it by the others can produce the individual RFPs. For the iris dataset (Figure 14) the individual RFPs (C_m) have been reordered by the known iris groups {(1) Setosa, (2) Vericolor, (3) Virginica}. In these matrices, high counts are represented with yellow and low counts with red. It is clear that the groups are differently expressed across each response, (more expressed groups have a high count, and appear more yellow in colour). The consensus matrix $\bar{C}$ is a mean representation over all individual matrices.

Figure 14: An illustration of combining RFPs into a consensus proximity matrix, $\bar{C}$, using the iris dataset.



In this thesis we propose three methods to estimate $\bar{C}$: Generalised Procrustes Analysis (GPA) (Gower, et al. 1996), a hierarchical beta binomial model (BB) (Gelman, et al. 1997) and Plaid Models (PLAID) (Lazzeroni, et al. 2002).

GPA looks for an average principal component representation over all C_m. Principal components analysis (PCA) maps the correlation structure within the RFPs in a reduced dimension space. Therefore, finding an average PCA will highlight the dominant correlation structure.

A hierarchical beta binomial model observes an individual count, c_{ijm}, over the M RFPs. The model assumes that each count follows a binomial distribution with a probability parameter, θ_{ijm}. These θ_{ijm}s are assumed to follow an overall beta distribution. The interpretation of this distribution is that it summarises the probabilities from each binomial distribution over all the M RFPs. From the beta binomial model the expectation of this distribution is estimation and taken as the overall estimate for the probability of c_{ij} in the consensus matrix. For a large number of counts (responses) the beta binomial model approaches a normal distribution mean, however for small numbers of counts it provides a robust estimate of the probability distribution (Gelman, et al. 1997).

The third method uses plaid models to find responses that are similar in the RFPs, and explicitly model those that are different. Plaid models for MCTs look for the most stable mean representation over the response RFPs. To do this, RFPs are weighted by their importance to the average representation. Those found to diverge from the mean are entered as parameters into the model and the degree of divergence is estimated.

Plaid models offer a way of identifying responses that do not display the consensus grouping structure.

3.5.1 Combining RFPs by general procrutes analysis (GPA)

Orthogonal General Procrustes Analysis (GPA) (Gower, et al. 1996), minimises the Euclidean norm,

$$\sum_{m=1}^{M} \left\| C_m Q_m - \bar{C} \right\| \tag{3.24}$$

where Q_m is an orthogonal rotation on C_m and $\bar{C}$ is the new mean configuration of $\{C_1,\ldots C_m,\ldots,C_M\}$. Before the algorithm starts, the global mean is subtracted from each C_m. After iteration *i*, the new global mean $\bar{C}^{i+1}$ is defined by,

$$\bar{C}^{i+1} = \bar{C}^{i} + \frac{1}{M}\sum_{m=1}^{M} C_m^i Q_m^i \tag{3.25}$$

The Q_m^i's are found by a Singular Value Decomposition (SVD) on C_m^i, representing it by $C_m^i = U_m^i \Sigma_m^i \left(V_m^i\right)^T$ where U_m^i and V_m^i are the matrices of the eigenvectors of C_m^i and its eigenvalues are stored in Σ_m^i. As, in this case C_m^i is a symmetric matrix each Q_m^i is found by:

$$Q_m^i = \sum_{k=1}^{n_k} \frac{1}{\sqrt{\lambda_k}} \left(U_k\right)_m^i \left(\left(U_k\right)_m^i\right)^T \tag{3.26}$$

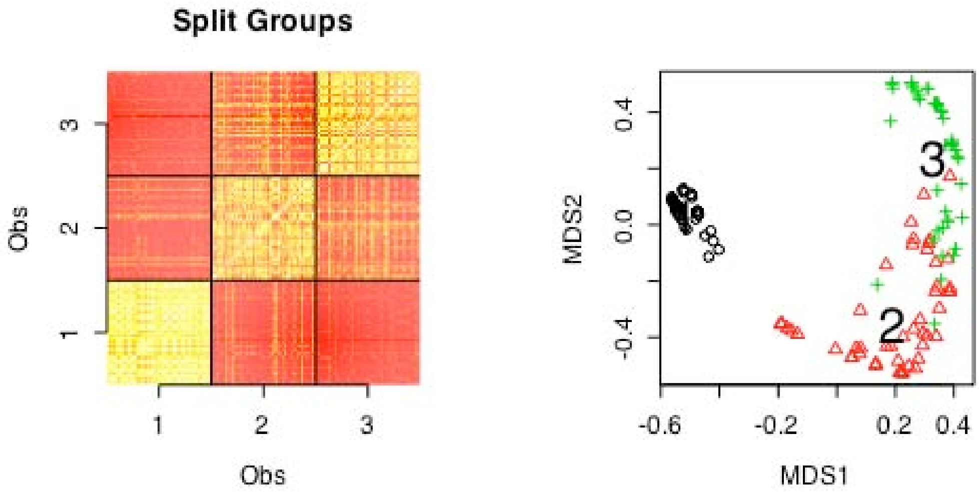
where the SVD is taken to n_k components, and U_k are the singular vectors with the largest corresponding eigenvectors λ_k. Once each Q_m is known the algorithm then updates $\bar{C}$ and each C_m is rotated in the direction of Q_m^i, redefining $C_m^{i+1} = C_m^i Q_m^i$.

By expanding (3.24) (Gower 1975), an error measurement of the new configuration can be attained:

$$error^{i}=\sum_{m=1}^{M}trace\left(\left(\bar{C}^{i}\right)^{T}\bar{C}^{i}+C_{m}^{i}\left(C_{m}^{i}\right)^{T}-2\bar{C}^{i}Q_{m}^{i}\left(C_{m}^{i}\right)^{T}\right) \qquad (3.27)$$

which is the reduction in residual sums of squares of the new configuration. The iterations stop when the error converges. The result is a overall representation of the structure of all matrices (Figure 15).

Figure 15: GPA consensus matrix for the iris dataset: (1) setosa, (2) versicolour, (3) virginica.



3.5.2 Combining RFPs by a hierarchical beta-binomial model (BB)

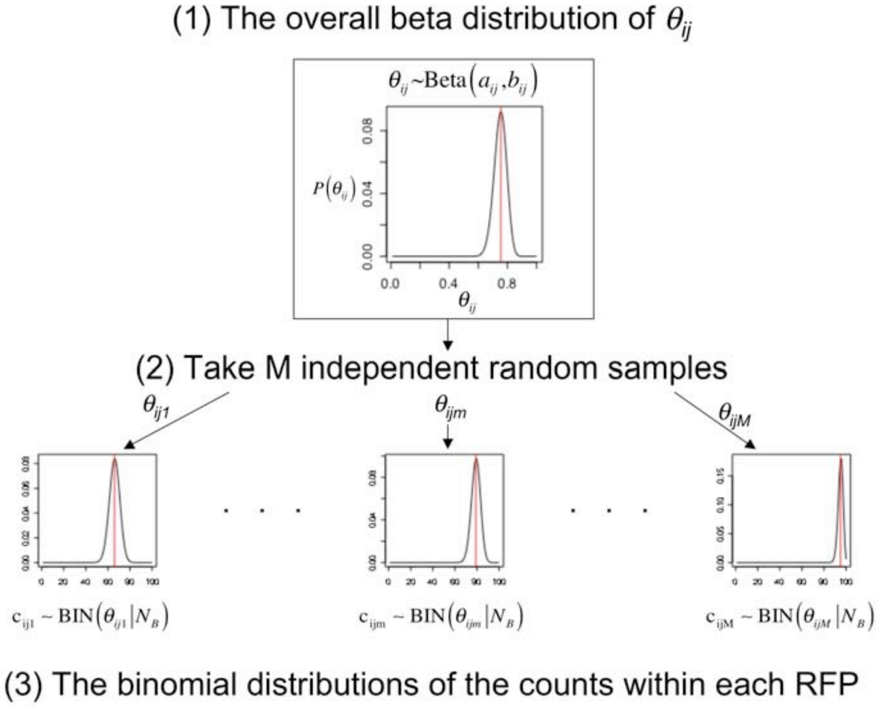
The beta binomial model considers each cell over all matrices individually. Each cell within a single RFP, c_{ijm}, represents the number of times the two observations i and j have been placed in the same terminal node within the ensemble to predict response variable m. Therefore c_{ijm} can be considered a binomial distribution,

$c_{ijm} \sim BIN\left(N_B, \theta_{ijm}\right)$ where θ_{ijm} is the binomial probability parameter that observations *i* and *j* are placed in the same terminal node of the ensemble consisting of N_B trees.

From each response variable's RFP, taking the same cell gives a vector of *M* counts $\underset{\sim}{c}_{ij} = \left\{c_{ij1}, \cdots, c_{ijm}, \cdots, c_{ijM}\right\}$. Each count, c_{ijm}, is assumed to have a binomial distribution. The goal of the analysis is to estimate an expected value over all counts, $\hat{c}_{ij} = E\left(\underset{\sim}{c}_{ij}\right)$, that is representative of all responses. To do this a hierarchical beta-binomial model is used (Gelman, et al. 1997).

The hierarchical beta-binomial model (Figure 16) assumes that the probability parameters that define each binomial distribution for each c_{ijm}, $\underset{\sim}{\theta}_{ij} = \left\{\theta_{ij1}, \ldots, \theta_{ijm}, \ldots, \theta_{ijM}\right\}$, are independent random samples from an overall distribution of the probabilities θ_{ij}. The overall distribution of the probabilities is assumed to be a beta distribution, $\theta_{ij} \sim Beta\left(a_{ij}, b_{ij}\right)$ with parameters a_{ij} and b_{ij}. This distribution is the generating distribution for the counts in $\underset{\sim}{c}_{ij}$.

Figure 16: Illustration of the hierarchical beta-binomial model.


(1) The overall beta distribution of θ_{ij}
θ_{ij} ~Beta(a_{ij}, b_{ij})
$P(\theta_{ij})$
0.00
0.04
0.08
0.0
0.4
0.8
θ_{ij}
(2) Take M independent random samples
θ_{ij1}
θ_{ijm}
θ_{ijM}
$c_{ij1} \sim \mathrm{BIN}(\theta_{ij1}|N_B)$
$c_{ijm} \sim \mathrm{BIN}(\theta_{ijm}|N_B)$
$c_{ijM} \sim \mathrm{BIN}(\theta_{ijM}|N_B)$
(3) The binomial distributions of the counts within each RFP

Estimating the expected value of the overall beta distribution, $\hat{\theta}_{ij} = E\left(\underset{\sim}{\theta}_{ij}\right)$, will give an overall probability that the observations have been placed in the same terminal node over all RFPs. The hierarchical Bayesian beta-binomial model estimates the hyperparameters a_{ij} and b_{ij} for the overall beta distribution of θ_{ij} .

3.5.2.1 Estimating the overall beta distribution

By Bayes' theorem, the joint posterior distribution over all parameters of the beta binomial model as a product of the prior distribution of the hyperparameters, a beta likelihood for the probabilities and a binomial likelihood for the counts is defined to be:

$$\begin{aligned} p\left(\theta_{ij}, a_{ij}, b_{ij} \left| \underset{\sim}{c}_{ij}\right.\right) &\propto p\left(a_{ij}, b_{ij}\right) p\left(\underset{\sim}{\theta}_{ij} \left| a_{ij}, b_{ij}\right.\right) p\left(\underset{\sim}{c}_{ij} \left| \underset{\sim}{\theta}_{ij}, a_{ij}, b_{ij}\right.\right) \\ &= p\left(a_{ij}, b_{ij}\right) \prod_{m=1}^{M} \frac{\Gamma\left(a_{ij}+b_{ij}\right)}{\Gamma\left(a_{ij}\right)\Gamma\left(b_{ij}\right)} \theta_{ijm}^{a_{ij}-1}\left(1-\theta_{ijm}\right)^{b_{ij}-1} \prod_{m=1}^{M} \theta_{ijm}^{c_{ijm}-1}\left(1-\theta_{ijm}\right)^{N_B-c_{ijm}} . \end{aligned} \tag{3.28}$$

As there are only two hyperparameters that need to be estimated, their distribution $p\left(a_{ij}, b_{ij}\right)$ can be simulated directly using a contouring approach. The implementation within this thesis follows (Gelman, et al. 1997) closely. By assuming that each c_{ijm} is independent the joint density of $\underset{\sim}{\theta}_{ij}$ is the beta likelihood:

$$p\left(\underset{\sim}{\theta}_{ij} \left| a_{ij}, b_{ij}, \underset{\sim}{c}_{ij}\right.\right) = \prod_{m=1}^{M} \frac{\Gamma\left(a_{ij}+b_{ij}+N_B\right)}{\Gamma\left(a_{ij}+c_{ijm}\right)\Gamma\left(b_{ij}+N_B-c_{ijm}\right)} \theta_{ijm}^{a_{ij}+c_{ijm}-1}\left(1-\theta_{ijm}\right)^{b_{ij}+N_B-c_{ijm}-1} . \tag{3.29}$$

By using conditional probability,

$$p\left(a_{ij}, b_{ij} \left| \underset{\sim}{c}_{ij}\right.\right) = \frac{p\left(\theta_{ij}, a_{ij}, b_{ij} \left| \underset{\sim}{c}_{ij}\right.\right)}{p\left(\theta_{ij} \left| a_{ij}, b_{ij}, \underset{\sim}{c}_{ij}\right.\right)}, \tag{3.30}$$

the joint distribution of the hyperparameters can be extracted by dividing (3.28) by (3.29) to give,

$$p\left(a_{ij}, b_{ij} \left| \underset{\sim}{c}_{ij}\right.\right) = p\left(a_{ij}, b_{ij}\right) \prod_{m=1}^{M} \frac{\Gamma\left(a_{ij}+b_{ij}\right)}{\Gamma\left(a_{ij}\right)\Gamma\left(b_{ij}\right)} \frac{\Gamma\left(a_{ij}+c_{ijm}\right)\Gamma\left(b_{ij}+N_B-\underset{\sim}{c}_{ijm}\right)}{\Gamma\left(a_{ij}+b_{ij}+N_B\right)} . \tag{3.31}$$

Evaluating (3.31) over a suitable range will produce a contour plot of the joint density from which estimates of the hyperparameters can be attained. To simplify the computations the natural logarithm of (3.31) is used. The prior for the hyperparameters is specified as uniform over the range $p\left(\log\left(\frac{a_{ij}}{b_{ij}}\right), \log\left(a_{ij}+b_{ij}\right)\right)$ (Gelman, et al. 1997).

A contour run over the initial estimate of its range may result in the density in (3.29) not being completely encompassed. To prevent this, the boundaries of the contour are summed, if the sum of the probabilities on the boundaries is greater than 0.01 (or less than 99% of the distribution being modelled), the boundaries are extended and the contouring is repeated. Once the distribution has been sufficiently modelled, the values of the hyperparameters are extracted by reading off the maximum.

3.5.2.2 Estimating the overall count

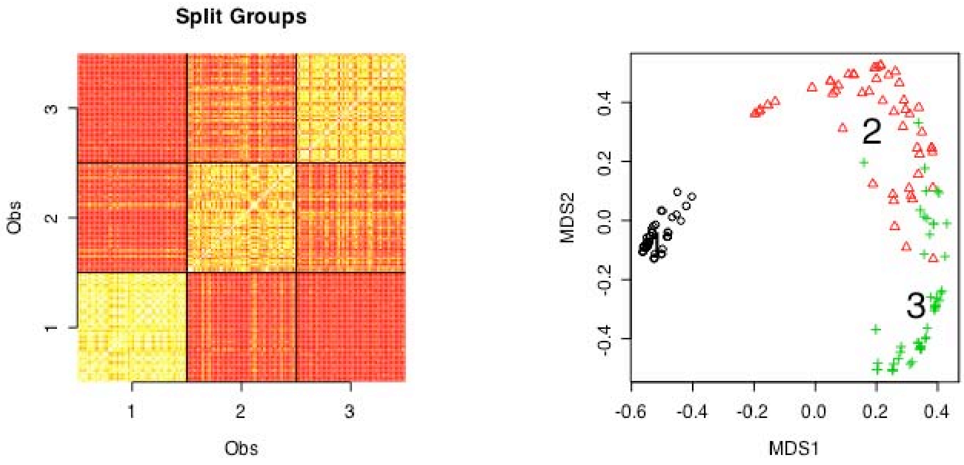
Now that the hyperparameters have been estimated and assuming a binomial distribution, a formula for the estimation of the overall count $\hat{c}_{ij}$ is found to be,

$$\hat{c}_{ij} = E\left(BIN\left(N_B, \theta_{ij}\right)\right) = N_B E\left(\theta_{ij}\right) = N_B\left(\frac{a_{ij}}{a_{ij} + b_{ij}}\right). \tag{3.32}$$

The result of this procedure is a robust estimation of the expected value of $\underset{\sim}{c}_{ij}$, where $\hat{c}_{ij}$ is the estimated count. These expectations are combined to form the overall consensus matrix $\bar{C} = \begin{bmatrix} \hat{c}_{11} & \vdots & \hat{c}_{1N} \\ \cdots & \hat{c}_{ij} & \cdots \\ \hat{c}_{N1} & \vdots & \hat{c}_{NN} \end{bmatrix}$.

For the iris dataset the overall consensus matrix is displayed in Figure 17.

Figure 17: BB consensus matrix for the iris dataset: (1) setosa, (2) versicolour, (3) virginica.



3.5.3 Combining RFPs by plaid models (PLAID)

Plaid models (Lazzeroni, et al. 2002) are a two-way clustering algorithm as they define a group as a subset of variables and observations. Developed initially for micro-array clustering, a plaid model searches for blocks of observations and variables that show a common pattern. Plaid models find one subset (layer) at a time in a forward stage-wise fashion. Each new layer is found in a similar way to adding a new model in boosted regression, as the effect of previous layers are subtracted, and the next layer is found on the residual data matrix.

As an RFP combination method, plaid models perform a search for a stable mean consensus matrix by grouping RFPs that have similar configurations. This method treats each RFP as a single variable in plaid models. To do this each RFP of (dimension *n* by *n*) is converted into a vector of length n^2 by,

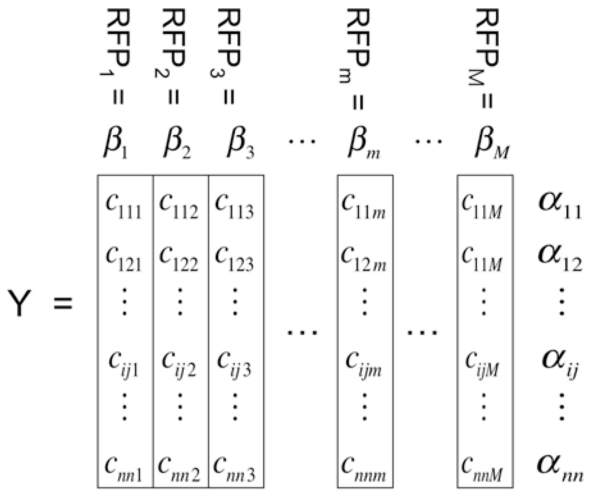
$$RFP = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & & \\ \vdots & & c_{ij} & \\ c_{n2} & & & c_{nn} \end{bmatrix} \xrightarrow{\text{is converted to}} \begin{bmatrix} c_{11} \\ \vdots \\ c_{1n} \\ c_{21} \\ \vdots \\ c_{2n} \\ c_{31} \\ \vdots \\ \\ c_{nn} \end{bmatrix} \xrightarrow{\text{is denoted by}} \underrightarrow{RFP} \qquad (3.33)$$

and then these vectors are concatenated into a data matrix,

$$Y = \begin{bmatrix} \vdots & \vdots & & \vdots \\ \underrightarrow{RFP_1} & \underrightarrow{RFP_2} & \cdots & \underrightarrow{RFP_M} \\ \vdots & \vdots & & \vdots \end{bmatrix}. \qquad (3.34)$$

Plaid models estimate the entire dataset using a sum of *K* layers, where each layer defines a homogeneous group within the data. A subset of observations and variables of the data matrix Y define each layer. These subsets are specified for each layer, *k*, by binary indicator variables, $\rho_{(ij)k}$ for the rows and κ_{mk} for the variables. A '1' in these vectors indicates that the structure within that observation or variable deviates from the mean of that layer, μ_k. The magnitudes of these deviations are estimated in the parameters, $\alpha_{(ij)k}$ for the observation effects and β_{mk} for the variable effects respectively. This results in two sets of parameters: ($\rho_{(ij)k}$, $\alpha_{(ij)k}$) which estimates how representative each observation is of μ_k and (κ_{mk}, β_{mk}) which estimates how representative each variable is of μ_k (Figure 18).

Figure 18: Plaid model illustration for a single layer.


RFP₁ =
RFP₂ =
RFP₃ =
RFPₘ =
RFPₘ =
β₁ β₂ β₃ ··· βₘ ··· βₘ
Y =
c₁₁₁ c₁₁₂ c₁₁₃ c₁₁ₘ c₁₁ₘ α₁₁
c₁₂₁ c₁₂₂ c₁₂₃ c₁₂ₘ c₁₁ₘ α₁₂
cᵢⱼ₁ cᵢⱼ₂ cᵢⱼ₃ cᵢⱼₘ cᵢⱼₘ αᵢⱼ
cₙₙ₁ cₙₙ₂ cₙₙ₃ cₙₙₘ cₙₙₘ αₙₙ

In this thesis, to construct the overall consensus matrix, plaid models are run on *Y* (Figure 18) to a single layer, *K = 1*, and therefore the *k* index can be dropped from the model. The mean representation of a single count between observations *i* and *j*, c_{ij}, can be found as a sum over the *M* RFPs,

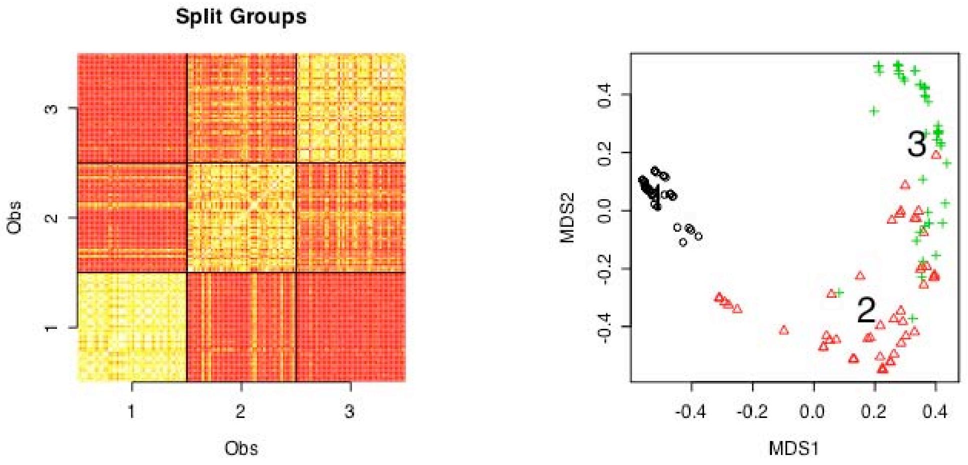
$$\hat{c}_{(ij)} = \mu_0 + \left(\mu_k + \rho_{(ij)k}\alpha_{(ij)k} + \sum_{m=1}^{M} \kappa_{mk}\beta_{mk} \right). \qquad (3.35)$$

Doing this for all counts will produce a consensus matrix of the form,

$$\bar{C} = \begin{bmatrix} \hat{c}_{11} & \hat{c}_{12} & \cdots & \hat{c}_{1n} \\ \hat{c}_{21} & \hat{c}_{22} & & \\ \vdots & & \hat{c}_{ij} & \\ \hat{c}_{n2} & & & \hat{c}_{nn} \end{bmatrix}. \qquad (3.36)$$

For the iris dataset the result of plaid combining is shown in Figure 19.

Figure 19: PLAID consensus matrix for the iris dataset: (1) setosa, (2) versicolour, (3) virginica.



3.5.3.1 *Estimating the plaid parameters*

Plaid models use a forward stage-wise addition of layers. For each element in *Y*, this is expressed as a linear combination of row and column effects over *K* layers,

$$c_{(ij)m} = \mu_0 + \sum_{k=1}^{K} \left(\mu_k + \rho_{(ij)k} \alpha_{(ij)k} + \kappa_{mk} \beta_{mk} \right), \tag{3.37}$$

where each new layer is found by estimating the parameters on the residuals from the previous layers.

For each layer, plaid models iteratively minimise the loss function *Q*,

$$Q_k = \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{m=1}^{M} \left(z_{ijmk} - \left(\mu_{(ij)k} + \rho_{(ij)k} \alpha_{(ij)k} + \kappa_{mk} \beta_{mk} \right) \right)^2, \tag{3.38}$$

where z_{ijmk} are the residuals from the previous layer. Plaid model estimate the parameters using the method of Lagrange multipliers, subject to a loose set of constraints,

$$0 = \sum_{i=1}^{N} \sum_{j=1}^{N} \rho_{(ij)k}^2 \alpha_{(ij)k} = \sum_{m=1}^{M} \kappa_{mk}^2 \beta_{mk}, \tag{3.39}$$

that forces the minimisation to use positive values of $\rho_{(ij)k}$ and κ_{mk}, but also have the effect of shrinking small values of these parameters close to zero. These constraints however do not enforce the necessary conditions that $\rho_{(ij)k} \in [0,1]$ and $\kappa_{mk} \in [0,1]$.

To enforce that $\rho_{(ij)k}$ and κ_{mk} lie within the range [0,1], the updates are never implemented. Instead *S* iterations are run and if the update for $\rho_{(ij)k}$ or κ_{mk} at iteration *s* is above 0.5, then its value is updated by *0.5 + s/(2S)* and if it is below 0.5 the update is then *0.5 – s/(2S)*. This enforces a binary value of 0 or 1 in the final iteration and that at each iteration $\rho_{(ij)k} \in [0,1]$ and $\kappa_{mk} \in [0,1]$. Once all $\rho_{(ij)k}$'s and κ_{mk}'s have been found, a new layer is defined and consequently a new cluster has been determined.

The result from the minimisation procedure is the following parameter update formulae:

$$\begin{aligned}
\mu_k &= \frac{\sum_{ij}\sum_m \rho_{(ij)k}\kappa_{mk}z_{ijmk}}{\left(\sum_{ij}\rho^2_{(ij)k}\right)\left(\sum_m \kappa^2_{mk}\right)} \\
\alpha_{(ij)k} &= \frac{\sum_m \left(z_{ijmk} - \mu_k\rho_{(ij)k}\kappa_{mk}\right)\kappa_{mk}}{\rho_{(ij)k}\sum_m \kappa^2_{mk}} \\
\beta_{mk} &= \frac{\sum_{ij}\left(z_{ijmk} - \mu_k\rho_{(ij)k}\kappa_{mk}\right)\rho_{(ij)k}}{\kappa_{mk}\sum_{ij} p^2_{(ij)k}}
\end{aligned} \tag{3.40}$$

and using $\theta_{(ij)mk} = \mu_k + \alpha_{(ij)k} + \beta_{mk}$,

$$\begin{aligned}
\rho_{(ij)k} &= \frac{\sum_m \theta_{(ij)mk}\kappa_{mk}z_{ijmk}}{\sum_m \theta^2_{(ij)mk}\kappa^2_{mk}} \\
\kappa_{mk} &= \frac{\sum_{ij}\theta_{(ij)mk}\rho_{(ij)k}z_{ijmk}}{\sum_{ij}\theta^2_{(ij)mk}\rho^2_{(ij)k}}
\end{aligned} . \tag{3.41}$$

The most important parameter resulting from plaid model combining is the values in κ_{mk}. If in (3.35) the value of κ_{mk} is found to be '1', it means that the structure in this RFP deviates sufficiently from the mean such that it is necessary to explicitly add it as a parameter within the model. The magnitude of the corresponding β_{mk} gives an indication of how different that structure is. It is possible for plaid models not to find any κ_{mk}'s to be '1'. In this case all of the counts are sufficiently modelled by a stable mean representation over the RFPs.

The other parameters, $\alpha_{(ij)k}$, $\rho_{(ij)k}$ are necessary for the running of plaid models, however in the context of RFP combination are difficult to understand as they refer to a specific $\underset{\sim}{c}_{ij}$, as in the beta-binomial combination method. Their values read just like the β_{mk}'s and κ_{mk}'s as if $\rho_{(ij)k}$ is set to one, then that count differs from the mean representation, with $\alpha_{(ij)k}$ being a measure of the size of the deviation. If it is found to be a significant effect in the plaid model, it means that the counts over the RFPs for observations *i* and *j* do not follow the pattern of the other counts found as modelled by the consensus matrix.

4. Multivariate Consensus Trees (MCT)

A Multivariate Consensus Tree is a clustering or profiling model capable of finding stable groups over mixed type datasets. MCTs are tree-based models that search for decision rules within the predictor set to partition a consensus matrix into homogeneous groups. To do this MCTs define five new splitting criteria designed to find group structure within a consensus matrix. Furthermore, MCTs extend the *RE* graphs of CART (Section 3.2.1) to provide a way to estimate the optimal number of groups within the dataset.

This thesis proposes two algorithms for building an MCT; Global MCTs and local MCTs. Global MCTs construct an overall consensus matrix spanning all observations, and recursively partition on this matrix to build the tree. Local MCTs build a new consensus matrix at each terminal node to evaluate each new split. As local MCTs re-construct the consensus matrix for each split it is expected that they are more accurate in determining the split points of the tree. On the other hand global MCTs always observe the constant consensus matrix and therefore are likely to be adversely affected by competing group structures within the dataset.

4.1 MCT Splitting Functions

Splitting on a consensus matrix is very similar to splitting on a distance matrix as in Db-MRT (Figure 6). However with Db-MRT, the goal is to find the sub-matrices D_L and D_R containing small distances between the observations and D_C showing a large distance between the groups. As consensus matrices contain a similarity measure, the goal is to find group sub-matrices with high counts and a covariance sub-matrix with low counts. To not confuse Db-MRT with MCT methods, the consensus left and right sub-matrices will be denoted as S_L, S_R respectively and the covariance sub-matrix is denoted as S_C.

4.1.1 Splitting using sums of squares (SSR)

Sums of Squares Reduction Splitting (SSR) minimises the following,

$$R(d) = \sum_{i \in S_L} \sum_{j \in S_L} \left(c_{ij} - \bar{S}_L\right)^2 + \sum_{j \in S_R} \sum_{j \in S_R} \left(c_{ij} - \bar{S}_R\right)^2 + 2 \sum_{j \in S_C} \sum_{j \in S_C} \left(c_{ij} - \bar{S}_C\right)^2 \qquad (4.1)$$

where the group centroid is defined as the mean of all observations of a group. SSR splits are subject to a condition to ensure that a valid split is found. A valid split is defined when $\bar{S}_L > \bar{S}_C$ and $\bar{S}_R > \bar{S}_C$. This condition must be accepted before assessing the quality of a split; and states that the observations in the left and right groups of the split have been classified together more times than apart. As we are dealing with a similarity measure rather than a distance the validity condition is necessary to ensure a split results in meaningful nodes.

It is possible that no split on any variable in the dataset will meet the validity condition. If this happens the MCT cannot be grown any further. This could be because of two reasons:

1. There is no grouping structure within the data left to model. Therefore the terminal nodes are as pure as they will become.
2. The minimum terminal node size does not allow for the best split to be found.

4.1.2 Splitting using margin reduction (MR)

The margin of a classifier is defined as by how much the predictions made exceed random chance (Breiman 2001). When considering a split on an RFP, the best selected split should improve the grouping of objects. In terms of counts within a consensus matrix, this translates to having the counts within the terminal node sub-matrices of a potential split greater than the mean of the entire consensus matrix, $\bar{S}_T$. By this it is possible to define a correct and incorrect grouping for an observation c_{ij} within a terminal node sub-matrix:

1. A correct grouping of an observation is when $c_{ij} > \bar{S}_T$.
2. An incorrect grouping of an observation is when $c_{ij} \leq \bar{S}_T$.

Using these rules an impurity measure can be derived which maximises the ratio of correct and incorrect grouping observations,

$$R(d) = \frac{\left(\sum\limits_{i \in S_L} \sum\limits_{\substack{j \in S_L \\ c_{ij} \leq \bar{S}_T}} \left(c_{ij} \right) \right)}{\left(\sum\limits_{i \in S_L} \sum\limits_{\substack{j \in S_L \\ c_{ij} > \bar{S}_T}} \left(c_{ij} \right) \right)} + \frac{\left(\sum\limits_{i \in S_R} \sum\limits_{\substack{j \in S_R \\ c_{ij} \leq \bar{S}_T}} \left(c_{ij} \right) \right)}{\left(\sum\limits_{i \in S_R} \sum\limits_{\substack{j \in S_R \\ c_{ij} > \bar{S}_T}} \left(c_{ij} \right) \right)}. \tag{4.2}$$

If the sum of the incorrect groupings is small for both the left and right terminal nodes then (4.2) will decrease. If (4.2) decreases, the observations in the new left and right partitions have higher counts than the overall mean of all observations and the margin is maximised. Therefore by decreasing (4.2) the mean count in the new partitions is higher and the partition will increase the margin and the split is good.

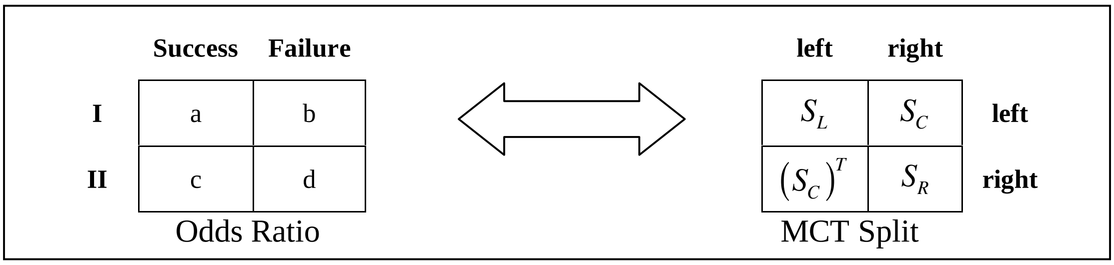
However using (4.2) it is possible to get an impurity of zero or an undefined impurity depending on the structure within the consensus matrix. An undefined impurity is a problem as this means that the sum of the correct counts is zero, and there is no valid split. Conversely a margin of zero is not a problem as this means that the sum of the incorrect counts is zero. This means that a very good split has been found where no incorrect groupings have been induced by the partition.

4.1.3 Splitting using an odds ratio (OR)

The odds ratio (Schork and Remington 2000) is a very common statistical tool for summarising the structure within two-way contingency tables (Figure 21),

$$OR = \frac{\text{Probability of Success}}{\text{Probability of Failure}} = \frac{ad}{bc}. \qquad (4.3)$$

Figure 20: MCT split as an odds ratio.



	Success	**Failure**
I	a	b
II	c	d

Odds Ratio

left	**right**	
S_L	S_C	**left**
$(S_C)^T$	S_R	**right**

MCT Split

It comes into use in MCTs because a partition on a consensus matrix can be treated as a two way contingency table,

$$OR = \frac{\text{Odds left}}{\text{Odds right}} = \frac{ad}{bc} = \frac{\left(\sum_{i \in S_L} \sum_{j \in S_L} \left(c_{ij}\right)\right)\left(\sum_{i \in S_R} \sum_{j \in S_R} \left(c_{ij}\right)\right)}{\left(\sum_{i \in S_C} \sum_{j \in S_C} \left(c_{ij}\right)\right)^2}. \tag{4.4}$$

As MCTs reduce impurity, the inverse of (4.4) is used to pick the best split. If the inverse of (4.4) is less than one, the sum of the left and right partitions is larger than that of the covariance partition and the split is good. If it is one or more, then the sums are the same, or the covariance sum is larger than the left and right partitions sum and the split is invalid.

4.1.4 Combining splitting functions (MR-SSR & OR-SSR)

One major advantage of the SSR split method is that it takes into account the group variation, however does not make use of the count structure of the data. By only considering the group variation, SSR may embed smaller groups within larger groups. The other techniques, (MR and OR) use only the count structure of the data, and do not estimate the variance of the group. By not considering the group variance it is possible that these methods may be biased towards smaller groups. To overcome these problems a combination of SSR with MR and OR to create two new splitting functions is proposed.

Combining SSR with MR (MR-SSR) gives the following impurity function:

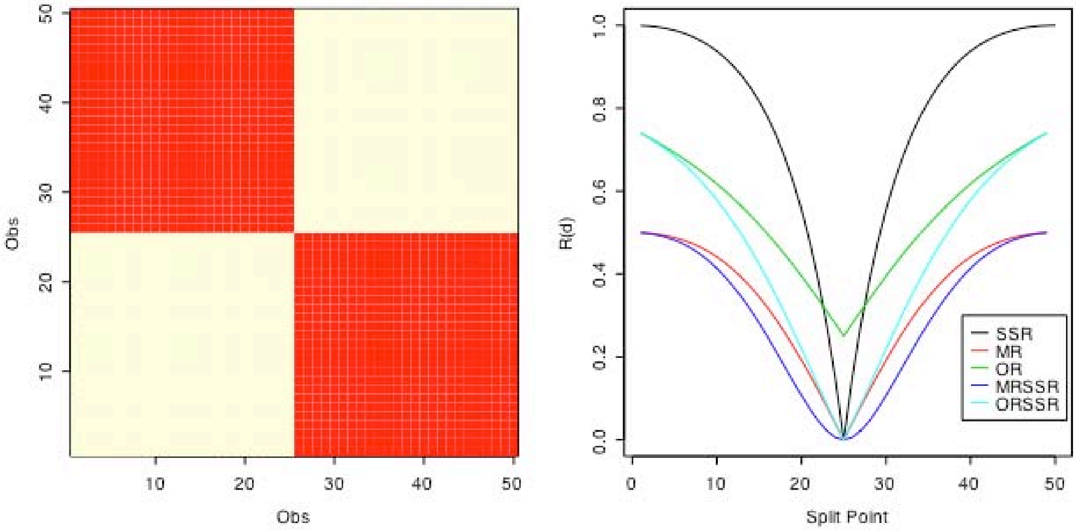
$$R(d) = MR(d) \times SSR(d) \tag{4.5}$$

Combining SSR with OR (OR-SSR) gives:

$$R(d) = OR(d) \times SSR(d) \,. \qquad (4.6)$$

Combination splitting methods are intended to remove split bias and improve performance. By weighting the count based rules by SSR, the path toward the best split becomes less steep, which increases the stability of the final split point (Figure 21).

Figure 21: Illustrating the performance of each individual MCT splitting function, on the case of perfect separation between the groups.



4.2 Growing An MCT

MCTs use a very general rule to pick the best split over many terminal nodes. Standard CART does this by picking the minimum of the RE statistic over all the nodes. MCTs can be more efficient as they have more information in the consensus matrix to assess the quality of a split, in particular the covariance between two nodes. The overall goal of an MCT split is to maximise the mean of counts of the new terminal nodes. This is done within each terminal node separately by finding the minimum of the impurity function.

To pick the next node upon which to split on, MCTs search for the best reordering of the consensus matrix into a block diagonal. Using the value of the impurity function to do this is likely to be biased toward picking terminal nodes with more observations. To find the next terminal node to grow on MCTs search over the best splits in all terminal nodes for the smallest $\overline{S}_C$. This finds the next two groups that are most well defined and gives the best re-ordering of the consensus matrix into a block diagonal form. Therefore it forces MCTs to identify the most clearly separated groups early in the tree and is unaffected by the specific splitting function.

4.3 Global MCTs

Global MCTs produce an overall consensus matrix and then grow the MCT on this matrix. The algorithm (Figure 22) relies heavily on the performance of the original random forests. The advantage of this approach is speed and that it allows for structure within the overall consensus matrix to be viewed with an MDS plot. Furthermore it is possible to directly observe the performance of each random forest, which can give an indication of response variable importance.

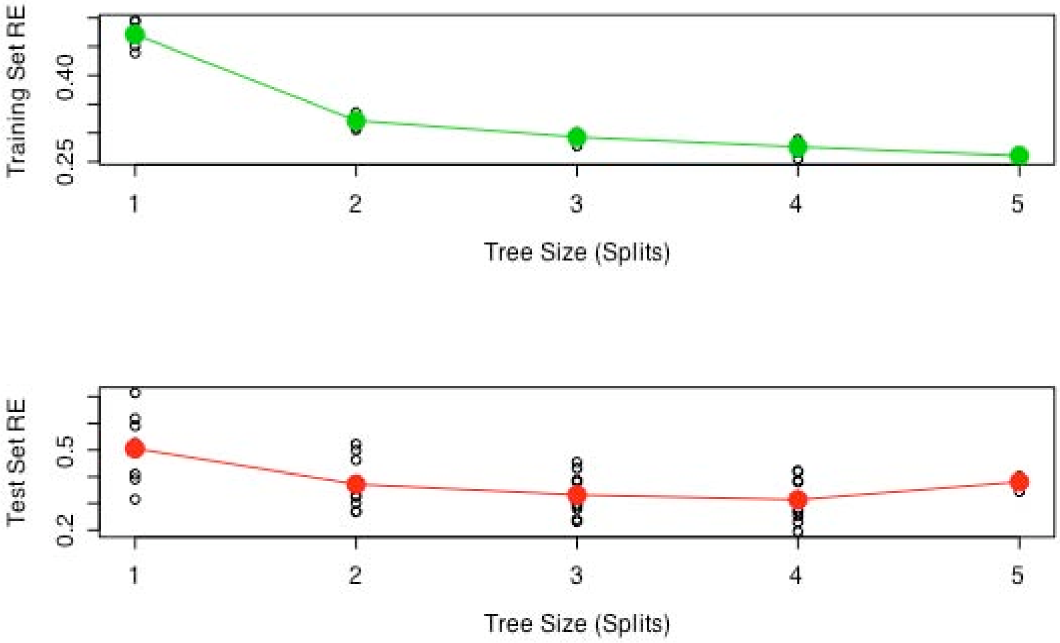
Figure 22: Global MCT algorithm.

```
1. For each response variable:
      a. Grow the forest and produce the RFP.
2. Produce the consensus matrix over all RFPs from each response matrix.
3. While tree size < maximum tree size do:
      a. Find the best split on each terminal node:
            i. For each predictor variable find the best split, $d$, by finding the
               minimum $R(d)$.
           ii. Over each predictor compare the best splits, and pick the
               variable with the smallest $R(d)$.
      b. For each terminal node compute the $RE(d)$ for the tree if that node was
         used to grow the tree.
      c. Compare the $RE(d)$ statistics over each terminal node and grow the tree
         on the minimum.
```

4.3.1 Tree size selection for global MCTs

As the response set for global MCTs is constant, the tree size can be estimated by V-fold cross validation as in standard CART. The RE statistic is defined as the sums of squares reduction as in SSR splitting (4.1). Over the course of the validation the left out observations are predicted by the centroid of the group in which they fall. The relative error statistic is computed on each test and training set in the validation. Finding the elbow in the RE curve (Figure 23) gives an estimate of the appropriate tree size to use. In the case of the iris dataset this is two splits or three terminal nodes (Figure 24).

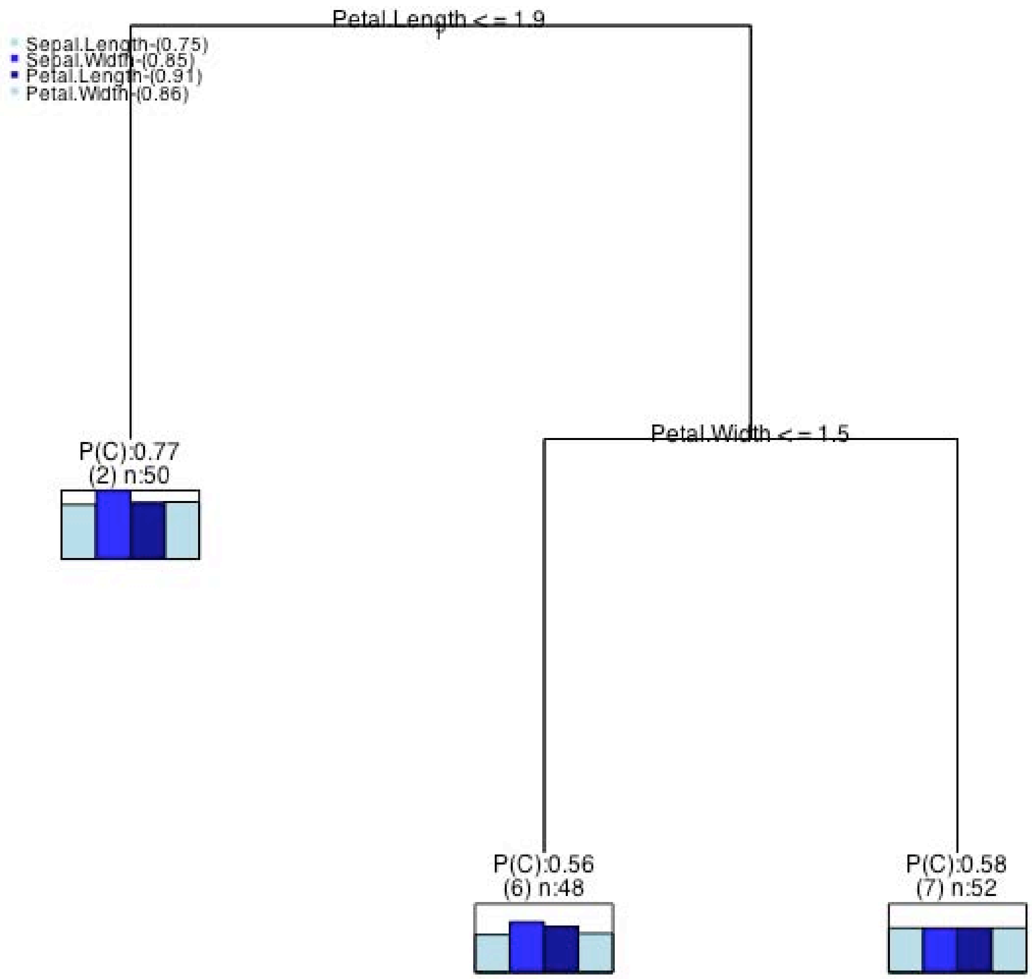
Figure 23: Global MCT 10-Fold CV for the iris dataset.



By predicting the consensus matrix in the cross-validation, global MCTs are finding the most stable number of clusters. The idea is very similar to other cross-validation regimes to determine the optimal number of groups (Dudoit, et al. 2002) and assessing the accuracy of a clustering solution (Tibshirani, et al. 2005).

For the iris dataset, based on the RE curve (Figure 23) the global MCT (Figure 24) is grown to 2 splits using SSR splitting and the GPA consensus matrix. Using the terminal node locations as groups this tree misclassifies 8 observations when compared with the known iris groups. A classification tree on the iris dataset grown to 3 terminal nodes misclassifies 6 observations.

Figure 24: Global MCT for the iris dataset.



4.4 Local MCTs

Local MCTs construct a new independent consensus matrix for the observations at each terminal node (Figure 25). In doing this they look for local grouping structure within a node, unaffected by the grouping structure in other nodes. By producing a consensus at each node the resolution of the groups is improved, and hopefully so too is the clustering accuracy. As local MCTs construct consensus matrices at each terminal node, they are computationally expensive, and do not allow for the analysis of the overall random forests to assess response variable importance.

Figure 25: Local MCT algorithm.

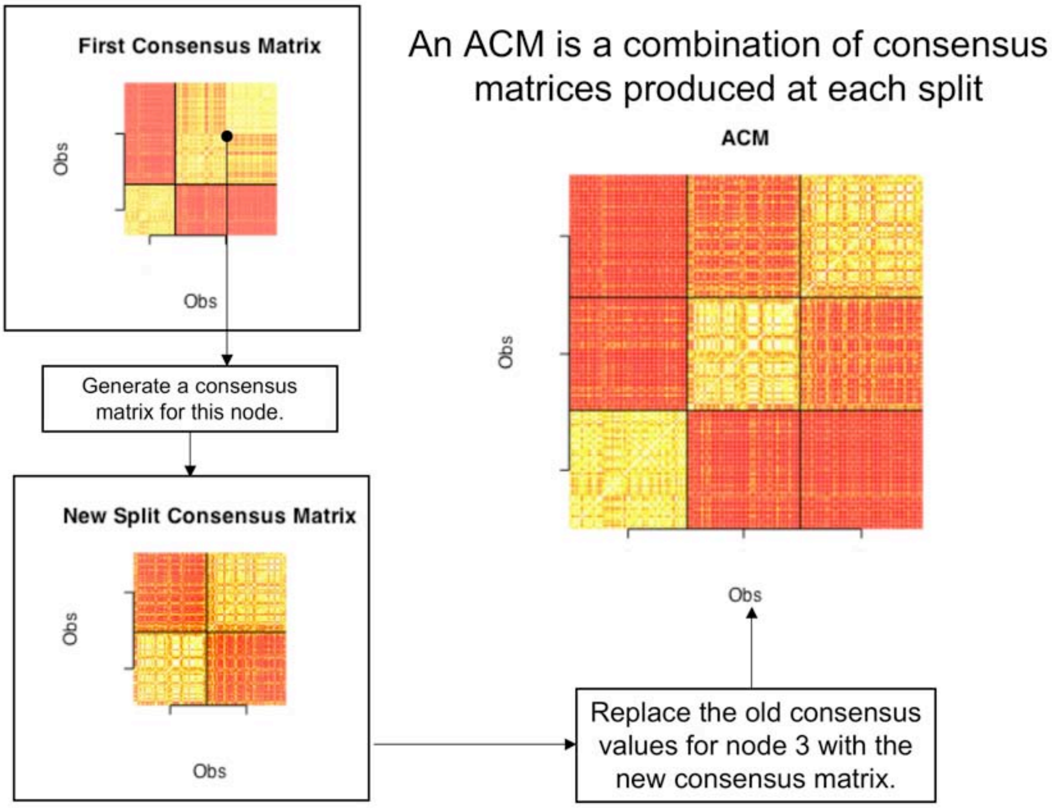
```
1. While tree size < maximum tree size do:
      a. Find the best split on each terminal node:
            i. For each response variable:
                  (a) Grow the forest and produce the RFP.
           ii. Produce the consensus matrix over all RFPs from each response
               matrix.
            i. For each predictor variable find the best split, $d$, by finding the
               minimum $R(d)$.
           ii. Over each predictor compare the best splits, and pick the
               variable with the smallest $R(d)$.
      b. For each terminal node compute the $RE(d)$ for the tree if that node was
         used to grow the tree.
      c. Compare the $RE(d)$ statistics over each terminal node and grow the tree
         on the minimum.
      d. Compute the augmented consensus matrix for that tree.
```

To allow local MCTs to produce a consensus matrix over all the observations they augment the individual consensus matrices for each split within the tree together into one overall consensus matrix. To do this local MCTs simply replace the areas in the old consensus matrix with the newly formed consensus matrix. This new matrix is the called the “augmented consensus matrix” (ACM).

4.4.1 The local MCT augmented consensus matrix (ACM)

An example construction of the augmented consensus matrix (ACM) is presented in Figure 26. Here, the first consensus matrix is constructed using all the available observations, as in global MCTs. A single partition is then made upon this matrix. After this partition, for each terminal node a separate consensus matrix is built using only the observations within the node. The best partition is then found for these matrices separately. These partitions are then compared and the best is selected and used to grow the tree. To produce the ACM the intermediate consensus matrix is then used to update the first consensus matrix.

Figure 26: Illustration of the construction of an ACM for the iris dataset.


First Consensus Matrix
Obs
Obs
Generate a consensus matrix for this node.
New Split Consensus Matrix
Obs
Obs
An ACM is a combination of consensus matrices produced at each split
ACM
Obs
Obs
Replace the old consensus values for node 3 with the new consensus matrix.

The disadvantages of local MCTs are the run time, and difficulty in summarising and monitoring the performances of each individual forest. More so, with local MCTs it is not possible to implement the standard V-fold cross-validation for model selection as the response consensus matrix changes with the addition of each new tree. To overcome this, an AIC statistic is produced to assist in model selection for local MCTs.

4.4.2 Tree size selection for local MCTs

As local MCTs cannot be cross-validated another method of model selection must be used that considers tree size. It is possible to get an indication of optimal tree size by using the Akaike Information Criterion (AIC) for a normal least squares problem (Burnham and Anderson 2002),

$$AIC = n\log\left(\hat{\sigma}^2\right) + 2K\ . \tag{4.7}$$

In a local MCT, each proximity c_{ij} of the ACM is grouped into a sub-matrix, S as a result of the partitions in the tree. If the groups within the MCT are stable, then the proximities within each sub-matrix should be well predicted by the centroid of that sub matrix $\overline{S}$,

$$ACM = \begin{bmatrix} S_1 & S_{12} & \cdots & S_{1T} \\ S_{21} & S_2 & & S_{2T} \\ \vdots & & \ddots & \\ S_{T1} & S_{T2} & & S_T \end{bmatrix} \sim \begin{bmatrix} \overline{S}_1 & \overline{S}_{12} & \cdots & \overline{S}_{1T} \\ \overline{S}_{21} & \overline{S}_2 & & \overline{S}_{2T} \\ \vdots & & \ddots & \\ \overline{S}_{T1} & \overline{S}_{T2} & & \overline{S}_T \end{bmatrix}. \tag{4.8}$$

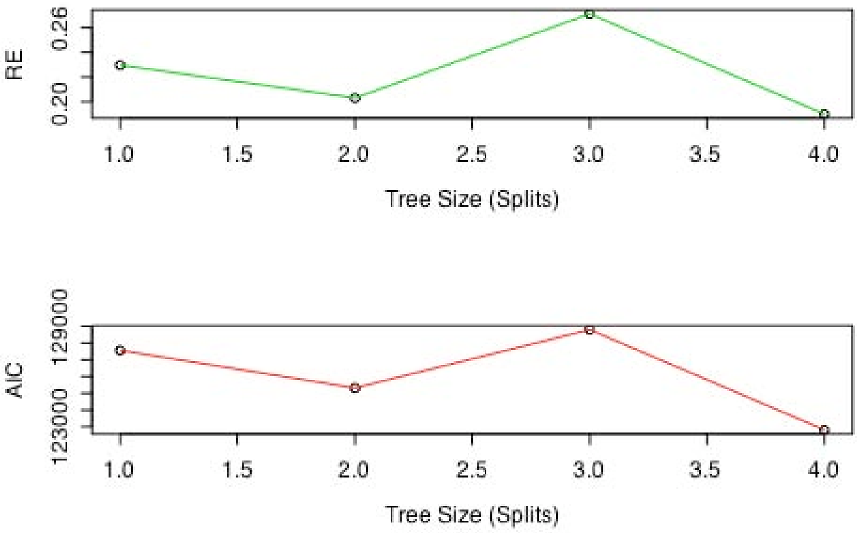
Therefore, by (4.7) and (4.8) the AIC for a local MCTs is defined to be,

$$AIC(T) = n^2 \log\left(\left(\frac{1}{n^2}\right)\sum_{i=1}^{n}\sum_{j=1}^{n}\left(c_{ij}^{ACM} - \overline{c}_{ij}^{ACM}\right)^2\right) + 4\left(\text{number of splits}\right) \tag{4.9}$$

where c_{ij}^{ACM} is a proximity within the ACM, $\overline{c}_{ij}^{ACM}$ is the centroid of its sub-matrix, and n is the number of observations within the dataset. The model size K, is 4 x (number of splits) as for each split there are 4 centroids to be estimated. By using the AIC it is possible to consider model size in selecting a local MCT model. The assumptions resulting from using the AIC for tree size selection is that the similarities within each sub-matrix of S follow a normal distribution and that the model complexity is that of a linear model with 4*(number of splits) parameters to be estimated.

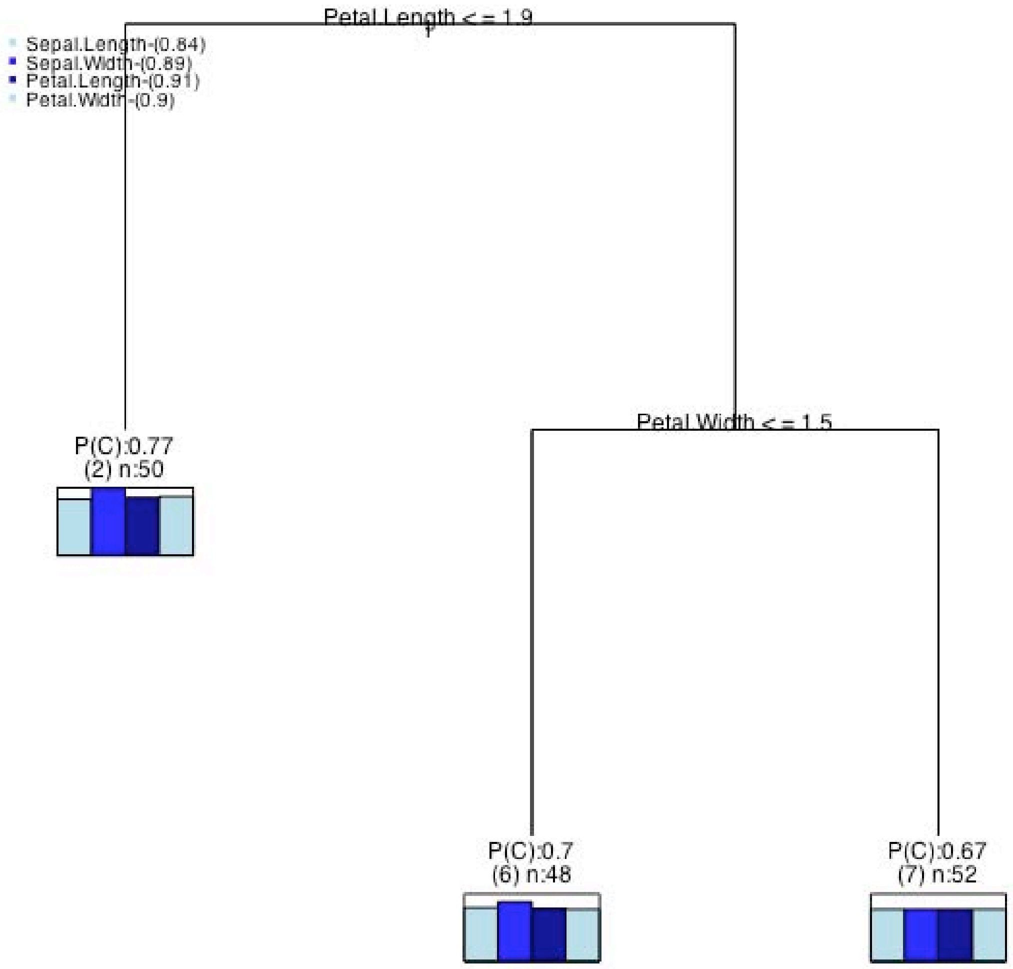
Unlike global MCTs the RE curve is not necessarily a decreasing function (Figure 27). This is because the ACM is continually updated, and in doing this it is not guaranteed that the RE will decrease. However this does not change the interpretation, as the optimal tree size still lies at the minimum, which in the case of the iris dataset is 2 splits or 3 terminal nodes (Figure 28).

Figure 27: Local MCT RE and AIC curves for the iris dataset.



For the iris dataset, based on the RE and AIC curves (Figure 27) the local MCT (Figure 28) is grown to 2 splits using SSR splitting and the GPA consensus matrix. Using the terminal node locations as groups this tree misclassifies 8 observations when compared with the known iris groups. A classification tree on the iris dataset grown to 3 terminal nodes misclassifies 6 observations.

Figure 28: Local MCT on the iris dataset.

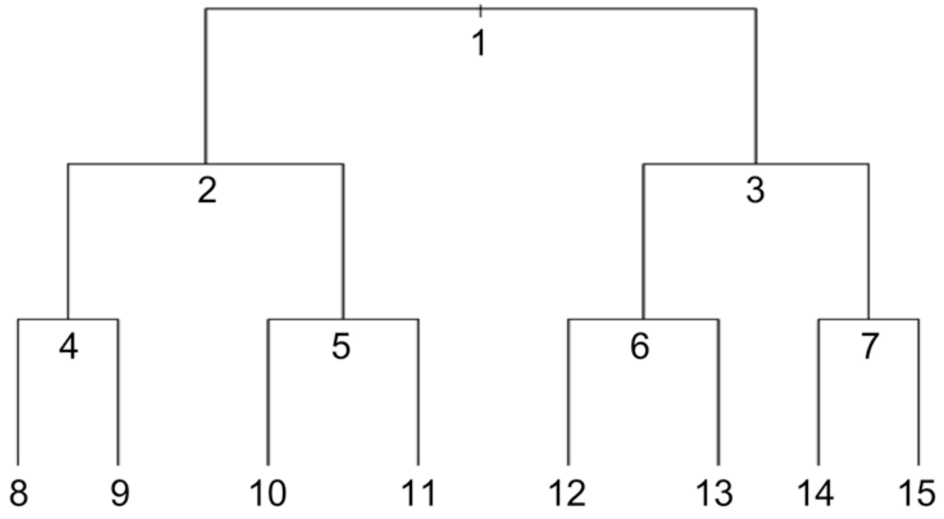


4.5 Understanding MCTs Output

4.5.1 Terminal node labelling

The numbering system used for the terminal nodes is simply that position in the entire tree from left to right (Figure 29). This provides a unique number for each possible terminal node location. Therefore there is no confusion as to where any terminal node is within the tree.

Figure 29: Terminal node numbering scheme.



4.5.2 Terminal node quality

Both local and global MCTs have some common statistics that help in the overall understanding of the trees (Figure 24, Figure 28). Firstly the centroid of each terminal node gives a measure of confidence in that node. Expressed in terms of a probability this measure is denoted as "*P(C)*" underneath each terminal node of the tree,

$$P(C) = \frac{\overline{S}_{Node}}{N_B} \tag{4.10}$$

where $\overline{S}_{Node}$ is the centroid for a node and N_B is the number of trees within the ensemble. The closer P(C) is to '1' the more times each observation within that terminal node has been positioned in the same node, and the more chance that node is a strong group within the data.

This information is also displayed for each individual response variable in the bar chart printed at each terminal node. In these charts, the longer the bar, the higher the expression of that node for that response variable. The order of the response variables in these charts is printed in the top left hand corner of the tree plot. These bar charts can be read like a variable importance list for the response variables for each terminal node.

Other information displayed on the tree is the node number, presented in brackets at each terminal node. Accompanying this is the number of observations within each terminal node.

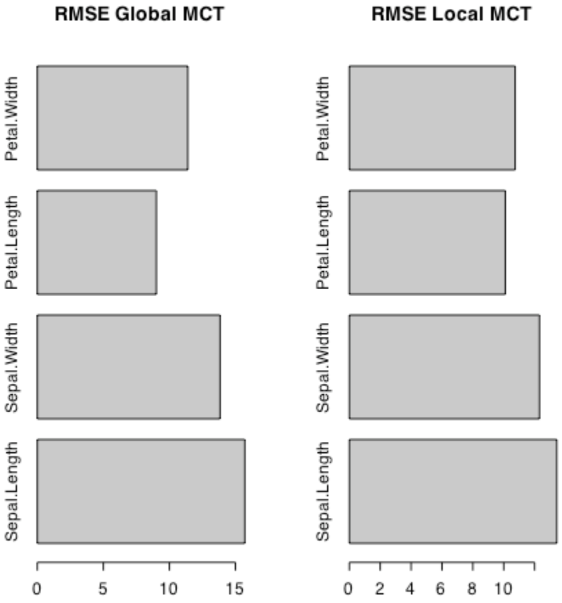
4.5.3 Assessing the quality of the consensus

After constructing the consensus matrix it is necessary to see how representative the result is. In this thesis a Root Mean Square Error (RMSE) is used,

$$RMSE\left(RFP_m\right)=\sqrt{\frac{\sum_{i=1}^{N}\sum_{j=1}^{N}\left(c_{ijm}-\hat{c}_{ij}\right)^2}{n^2}}\ . \tag{4.11}$$

The RMSE is used as it gives an error measurement in the form of a count, which allows for a more intuitive interpretation. For local MCTs, the RMSE is constructed between the ACM and the individual consensus matrices.

Figure 30: RMSEs for global and local MCTs for the iris example.



As can be seen for the iris dataset the RMSE profiles for local and global MCTs are the same (Figure 30). However it is clear that local MCTs have on average a lower RMSE than global MCTs. This is due to the updating of the consensus matrix with more accurate group structure.

4.5.4 Response variable importance (YVIP)

A response variable importance statistic for the entire MCT can also be computed. This is defined as an R^2 between the RFP for that response and a matrix of group centroids found by the tree:

$$YVIP\left(y_m\right) = 1 - \frac{\sum_{i=1}^{N}\sum_{j=1}^{N}\left(c_{ijm} - \overline{c}_{ijm}\right)^2}{\sum_{i=1}^{N}\sum_{j=1}^{N}\left(c_{ijm} - \overline{S}_m\right)^2} \tag{4.12}$$

where $\overline{c}_{ijm}$ is the centroid of the group in which c_{ij} has been placed, and $\overline{S}_m$ is the centroid of the entire RFP for response variable *m*. The YVIP of each variable is published in the MCT output (Figure 24, Figure 28) in the top left hand corner. The closer the YVIP is to '1', the more accurately the tree models that response variable.

4.5.5 Plaid terminal node filtering

The terminal nodes of an MCT correspond to sub-matrices that lie along the diagonal of the consensus matrix. At each terminal node MCTs assume that the counts within a sub-matrix can be modelled their mean centroid. This is also assuming that each terminal node sub-matrix for each response variables RFP are also sufficiently modelled by the mean centroid of the consensus matrix. This assumption in this thesis is called the "*homogeneity of a terminal node within an MCT*".

The consensus matrix of an MCT is a combination of RFPs for each response variable. As the combination methods are designed to find the dominant structure over all response variables, it is possible that within a terminal node some RFPs will not express the structure found in the MCT consensus matrix. If so this is a violation of the assumption of a homogeneous terminal node. Plaid terminal filtering uses the PLAID combining algorithm to identify those RFPs within each terminal node that deviate from this assumption.

Plaid filtering extracts from each response variables RFP the sub-matrices corresponding to each terminal node of the MCT. These sub-matrices are entered into the plaid model combination method, and an analysis of the κ_ms is performed. If a κ_m is ‘1’, then plaid models have identified that the structure within that RFPs sub-matrix differs from the stable mean representation found by the plaid model. Therefore it violates the assumption of a homogeneous terminal node. A measure of by how much that RFP differs from the mean is estimated by β_m. Those RFPs with κ_ms equal to zero agree with the mean representation of that node.

Table 1: Plaid terminal node filtering of the iris dataset. The values in the table are *(κ_m)* x *(β_m)*. A zero represents agreement with the mean representation.

	Local MCT			**Global MCT**		
	2	**6**	**7**	**2**	**6**	**7**
Sepal Length	**0**	**0**	**0**	**0**	-25.98	-0.1
Sepal Width	**0**	15.79	16.12	14.65	17.37	**0**
Petal Length	-0.50	**0**	**0**	**0**	8.9	**0**
Petal Width	0.47	-15.79	-16.05	-14.56	-0.29	**0**

For the iris example (Table 1) the results clearly show that the local MCTs produce a more accurate consensus as each terminal node has its more representative variables, denoted as a ‘0’. These values are representative of the mean structure of the terminal node as they are found by plaid models not be expressed differently from the mean configuration and indicated with a κ_m of ‘0’. For global MCTs terminal node 6 is not easily found by any variable as no RFP is found to match the consensus RFP.

The non-zero values in Table 1 are the β_m coefficients of the plaid model for that node. A positive value indicates over expression, or a comparatively higher count than the mean centroid for that RFP sub-matrix, and negative values represent under expression or a comparatively lower count than the consensus mean centroid. It should be noted that the sum of the β_ms should be close to zero, as they must follow a normal distribution. Therefore, for a terminal node if one β_m is over-expressed, for example 15.79 for sepal width in terminal node 6 for the local MCT, plaid models are forced to find counterweights that are equally under expressed, in this case sepal width which is underexpressed at -15.79.

For the iris dataset it is clear that local MCTs and global MCTs are finding a different mean structure. Global MCTs show that all RFPs express the mean representation in terminal node 7, indicating that this is a stable group. However no RFPs display terminal node 6, in particular sepal length, which is underexpressed by -25.98. Through observation of the RFP bar plots for terminal node 6 in the global MCT (Figure 24), it can be seen that sepal width and petal length show a higher expression than sepal length and petal width. This is what is seen in the filtering results. The high expression of sepal length and petal width forced the stable mean representation to a higher value, which the plaid model then counteracted by assigning higher weights to the less expressed variables. The conclusion is that terminal node 6 for global MCTs is not a stable group.

For local MCTs a lot of symmetry in the plaid model parameters is observed. The pattern of the plaid coefficients over all RFPs for each terminal node is two variables of 0, one positive and one negative. In fact this is characteristic of a stable pattern. It can be seen that for terminal node 6 and 7 plaid models have selected sepal length and petal length as the stable mean representation for the nodes. By observation of the local MCT terminal node bar plots for these nodes (Figure 28), it can be seen that sepal length and petal length have similar expression levels to all other RFPs across these nodes, and therefore are good selections for the stable mean.

For terminal node 2 the bar plot shows that sepal width is highly over-expressed. In this case plaid filtering selected sepal length and sepal width to construct the mean representation. This is a choice of a relatively over-expressed variable to construct the mean has the result of inducing a high mean variation between the mean of the terminal node and the RFPs. This variation is large enough to encompass the expression of petal length and petal width. Therefore the observed deviations of these variables are small. This result could be due to a possible outlier effect induced by the over-expressed sepal width for that node.

Plaid terminal node filtering in combination with terminal node bar plots provides a useful tool to assess the quality of the groups found by an MCT. Those nodes with a stable mean representation are likely to be strongly expressed groups within the dataset. However plaid models can also highlight subsets of response variables that strongly express a particular node, or do not express a node at all. With careful interpretation of the results from plaid filtering it is possible to gain an understanding into what response variables express which group and a measure of the stability of these groups.

5. Software

All methods in this thesis were implemented in the R package for statistical computing (R Development Team 2005). The other techniques used are found within add-on packages to R, and referenced below.

1. The code for hierarchical agglomeration (AGNES) and partitioning around medoids (PAM) is found in the R contributed package "*cluster*" (Maechler, Rousseeuw, Struyf, Hubert and Hornik 2006).
2. Multivariate regression trees are found in the "*mvpart*" R contributed package (Therneau, Atkinson, Ripley and De'ath 2004).
3. MCTs, random forests and treeboost are all implemented in the R package "*mct*" developed during this thesis (Hancock 2006). This package implements the following algorithms:
    a. Random forests for classification and regression.
    b. Multivariate random forests with binary substitution for categorical response variables.
    c. Multivariate treeboost with binary substitution for categorical response variables.
    d. Global and Local MCT with the following functions:
        i. GPA, BB and PLAID proximity matrix combination methods
        ii. SSR, MR, OR, MR-SSR and OR-SSR splitting functions.
        iii. Plaid model terminal node filtering.
        iv. Plaid model variable filtering (Section 7.3.3.3).

6. MCT Sensitivity Analysis

The aim of this sensitivity analysis is two-fold. Firstly the aim is to assess the accuracy of the MCT splitting criteria and RFP combination methods, and secondly to assess what effect different random forest parameters have on the final MCT solution. These tests focus on three key stages in the construction of an MCT and their influential parameters (Table 2).

Table 2: Table of important MCT parameters

Construction Stage	Important Parameters	Possible Effect
Stage 1 Construction of the RFPs	Random forest tree size	Determines the number of groups within each RFP.
	Random forest terminal node size	Determines the minimum group size possible.
Stage 2 Computing the consensus matrix	GPA	Determines the structure in the final consensus.
	BB	
	PLAID	
Stage 3 Growing the MCT	Local or global MCT	Will have an effect on the accuracy of the final solution.
	MCT splitting criteria	Determines group structure.
	MCT terminal node size	Determines the minimum group size to be found.
	MCT tree size	Determined by V-fold cross-validation or by AIC.

The first sets of experiments are simulation tests designed to gauge the performances of the RFP combination and MCT growing methods described in stages 2 and 3 in Table 2. These experiments simulate RFPs such that they have a known and well defined group structure. This is done to remove any effect that different random forest parameters may have. These simulated RFPs are then run through global MCTs, using each combination technique and splitting criteria over a range of MCT tree and terminal node sizes. Furthermore RE curves are generated for each experiment to assess the ability of MCTs to estimate the number of groups in the datasets.

The second experiment is designed to assess the effect of the random forest parameters on the construction of the RFPs in stage 1. These experiments run local and global MCTs using random forests with different tree sizes and terminal node sizes to assess what affect these parameters have on the final MCT solution.

6.1 Simulation Tests

The following simulation tests aim to assess the sensitivity of MCTs to the consensus generation methods, choice of splitting criteria, and MCT terminal node and tree sizes. To do this several RFPs are simulated with known groups. Each RFP is then randomly blurred to simulate noise within a response set. The RFP combination methods will then be run to uncover the original base groups from the blurred RFPs. These simulation tests are intended to assess the quality and robustness of the MCTs, in the face of random variation within the RFP structure. Simulating the RFPs directly allows for an unbiased assessment of the performance of the RFP combination and the MCT splitting methods, as they are independent of any performance bias that random forests may have to either classification or regression responses variables or any type of predictor variable.

The performance of MCTs will be compared with the performance of standard methods, using the MCT created consensus matrix as their input. The comparison methods are:

- Hierarchical agglomeration (AGNES) using average, complete and Wards linkage (Section 2.3.1).
- Partitioning around medoids (PAM) (Section 2.3.2).
- K-means (Section 2.3.2).

To assess the relative quality of the different RFP combination methods a RMSE between the consensus matrix and the original non-blurred RFP is used. Here the lower the RMSE between the original RFP and the consensus matrix, the better that consensus generation method has found the original group structure.

As MCTs require a predictor variable to form a split, each MCT will be grown with a single predictor variable, which will be an index of the observations, where the known groups will be ordered along this index. As the index is the predictor variable it is possible for MCTs to anywhere split along its range. This knowledge of the order of the groups could give MCTs a distinct advantage over the comparison methods.

6.1.1 Simulating RFPs

To ensure that the exact group structure within each RFP is known, they are directly generated for a specified base group structure. The base configuration is a matrix that represents the centroids of each group within the RFP. From these base centroids the counts within the RFPs are samples taken from a uniform distribution centred about centroid that defines the group. To ensure sufficient within group variability exists, these counts are generated within a range defined one binomial standard deviation either side of the specified centroid.

For example, if the user specified groups centroids for a two group RFP is,

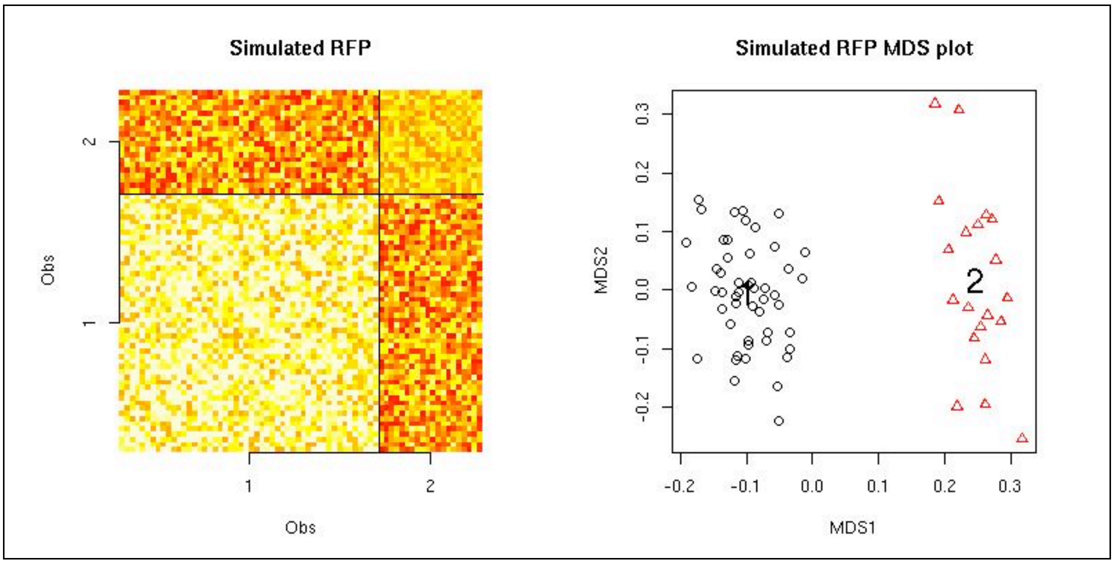
$$\begin{bmatrix} \bar{S}_L & \bar{S}_C \\ \left(\bar{S}_C\right)^T & \bar{S}_R \end{bmatrix} = \begin{bmatrix} 45 & 30 \\ 30 & 35 \end{bmatrix}$$

where the maximum count is $N_B = 50$ then the binomial standard deviation for each cell is defined as, $s = \sqrt{N_b p(1-p)}$, where p is the probability for a sub-matrix, given N_b. Given this, uniform random counts representing each group will be generated over the following domain:

$$\begin{bmatrix} 45 \pm \sqrt{4.5} & 30 \pm \sqrt{12} \\ 30 \pm \sqrt{12} & 35 \pm \sqrt{10.5} \end{bmatrix}.$$

If the group sizes are defined to be of 50 and 20 observations each, and a random seed is set at 1234567 the RFP and MDS plots in Figure 31 are generated. The RFP for group 1 comprises of 50 by 50 simulated counts, group 2 has 20 by 20 simulated counts and the covariate groups has 50 by 20 simulated counts.

Figure 31: Example of a simulated RFP.



Once the base RFP has been generated, to make the simulations more realistic, a random number from a uniform distribution within a specified range is randomly added or subtracted to each count within the RFP. This decreases the resolution of the groups, making them harder to find. At all stages of the simulations, it is ensured that the symmetry of the RFP is maintained and that the individual counts all lie between 0 and N_B.

6.1.2 Simulation Test 1: Four blurred equal sized groups

This experiment generates four blurry, but clearly separated groups. The groups here are equally sized to remove any outlying effects for the combination methods. The simulation parameters are:

- The random seed is initially set at 1234567.
- The group size for each group is 100 observations.
- The maximum count for any element in the RFP is 50.
- The original base configuration is in Table 3.
- Six blurred configurations are generated, by randomly adding or subtracting uniform random numbers between 0 and {5,10,15,20,25,30} to the original RFP. In the MDS plots of these RFPs (Figure 33) this is denoted by a +/- in the title.

Table 3: Four group simulation experiment base configuration group centroids.

		Group			
		1	**2**	**3**	**4**
Group	**1**	39	15	20	15
	2	15	30	15	10
	3	20	15	39	20
	4	15	10	20	35

The original configuration (Figure 32) shows the four simulated groups are quite close together with groups 2 and 3 slightly overlapping. The image of the RFP shows that the expressions of the groups are not that much different from the background expression. Once blurred, the groups become less obvious, with original +/- {15 (c), 25 (d) and 30 (e)} counts (Figure 33) not obviously showing all of the known groups.

Figure 32: Four group simulation base configuration.

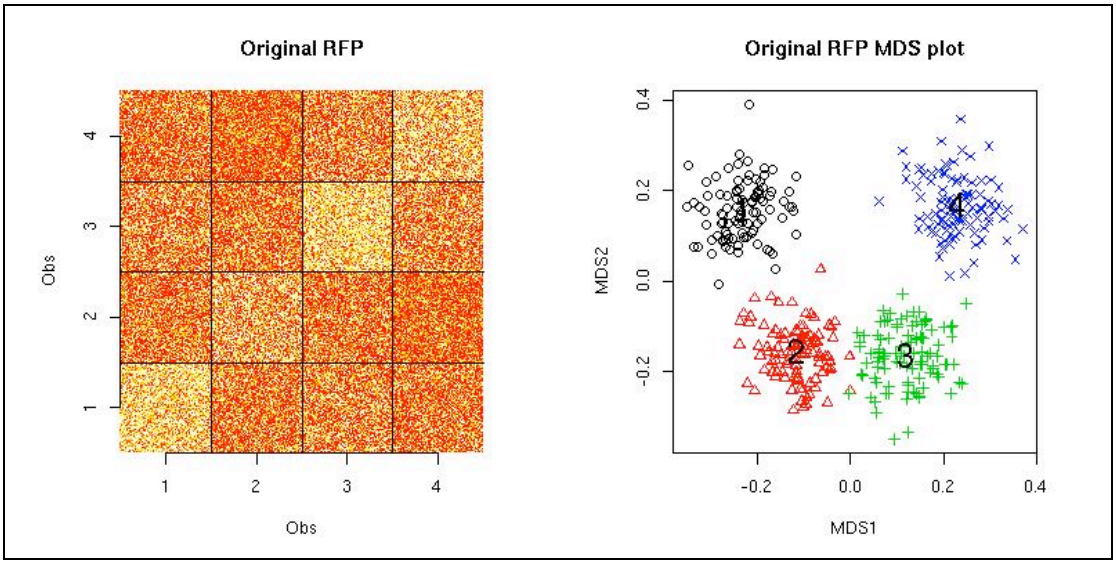
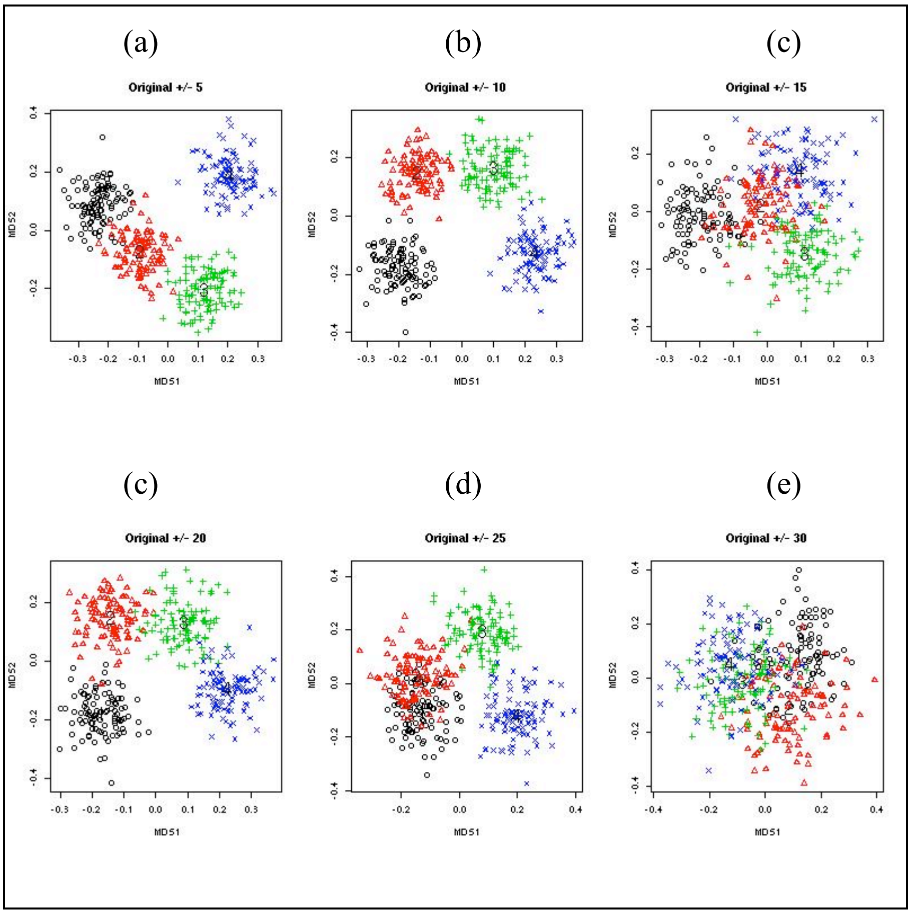
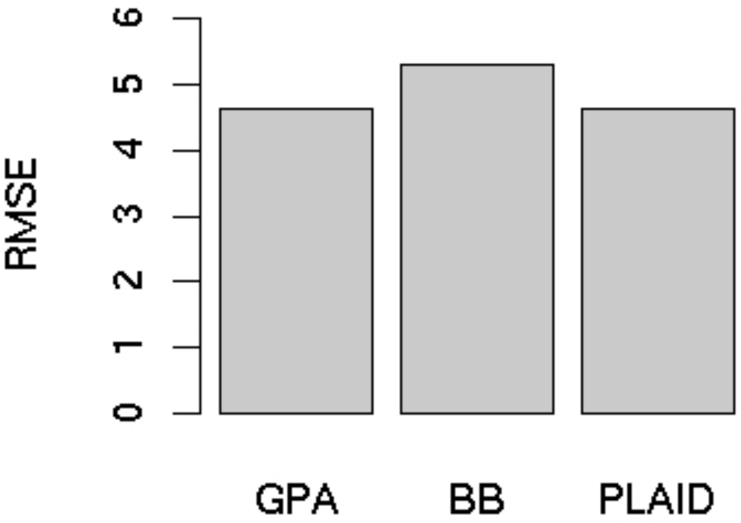


Figure 33: Four group simulation blurred RFP MDS plots.



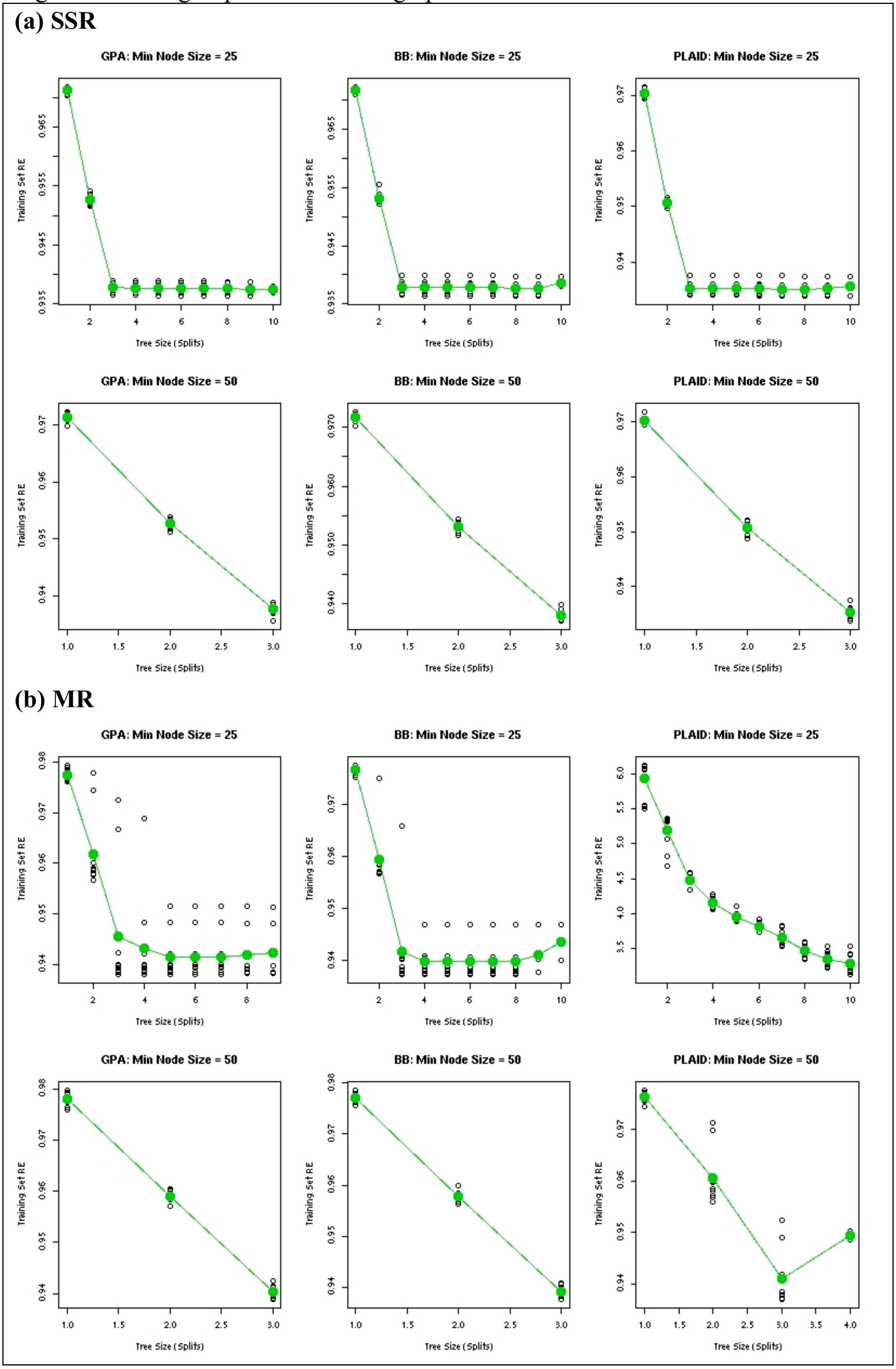
The combination methods are now run on the blurred configurations, without knowledge of the original base configuration. Of the combination methods it is clear from the RMSE plot that GPA and PLAID are performing equally best (Figure 34) and BB the worst. When the consensus configurations are compared side by side (Figure 36) there is no observable difference between the resulting MDS plots. Overall despite a weak original structure with additional blurring, the RMSE show that the combination methods only differed from the original by approximately 10 % (Figure 34).

Figure 34: Four group simulation combination RMSE.



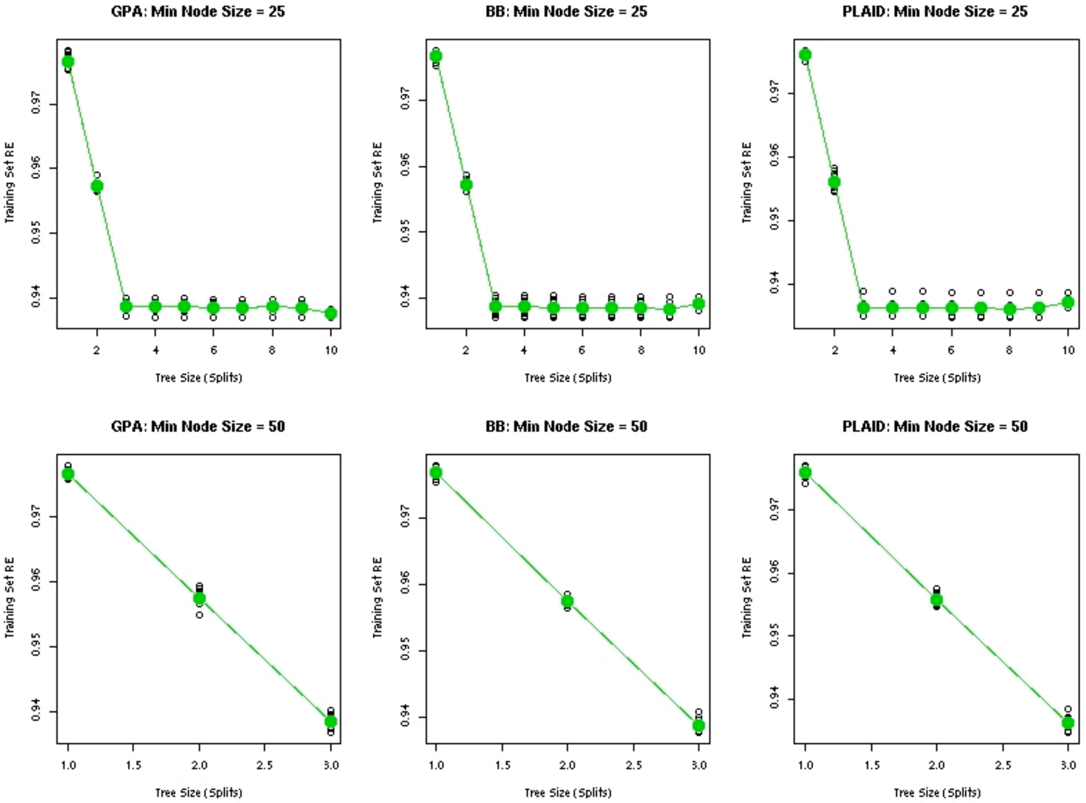
Over the different combination splitting methods, the RE curves for the same terminal node size show a consistent pattern (Figure 35). The known group size is 100 observations for each group. For the terminal node size set at 25 observations, all but the MR splitting function with the PLAID combining clearly show the elbow in the RE curves at 3 splits. When the terminal node size is increased to 50 observations, MCTs do not grow trees past 3 splits, again with the exception of MR splitting with PLAID combining which grew to 4 splits.

Figure 35: Four group simulation RE graphs.

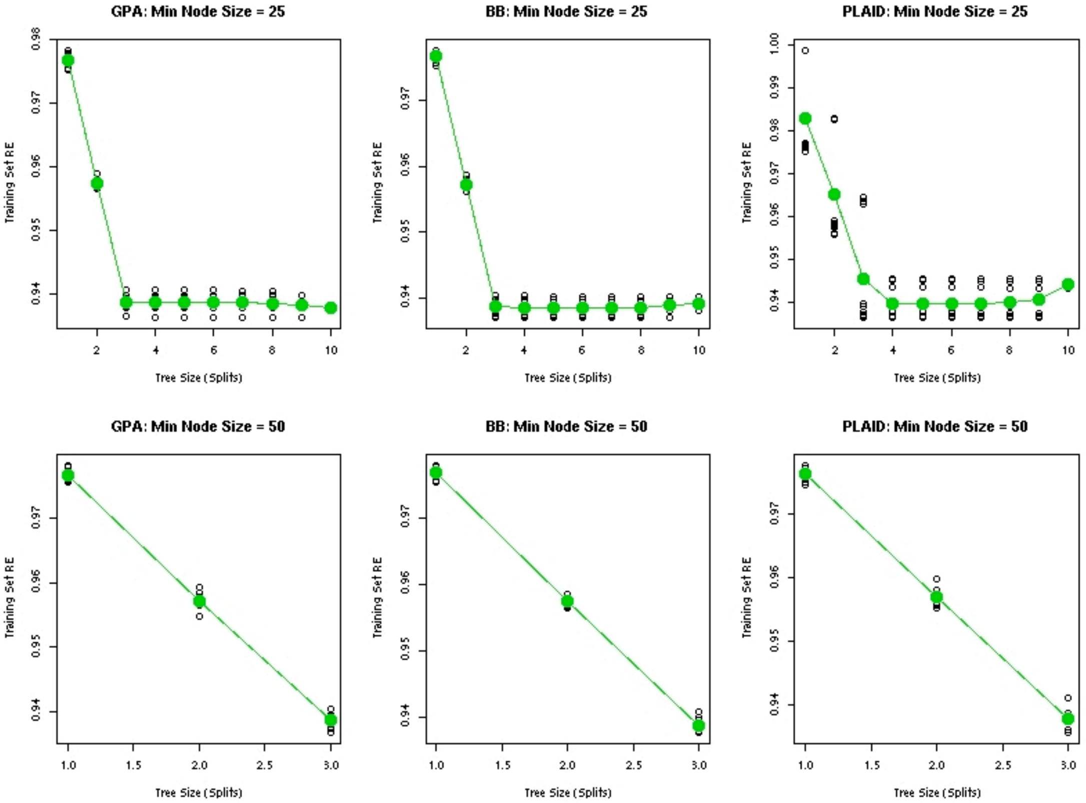

(a) SSR

(b) MR

(c) OR


GPA: Min Node Size = 25
BB: Min Node Size = 25
PLAID: Min Node Size = 25
GPA: Min Node Size = 50
BB: Min Node Size = 50
PLAID: Min Node Size = 50
Training Set RE
0.94
0.95
0.96
0.97
Tree Size (Splits)
2
4
6
8
10
1.0
1.5
2.0
2.5
3.0

(d) MR-SSR


GPA: Min Node Size = 25
BB: Min Node Size = 25
PLAID: Min Node Size = 25
GPA: Min Node Size = 50
BB: Min Node Size = 50
PLAID: Min Node Size = 50
Training Set RE
0.94
0.95
0.96
0.97
0.98
0.99
1.00
Tree Size (Splits)
2
4
6
8
10
1.0
1.5
2.0
2.5
3.0

(e) OR-SSR

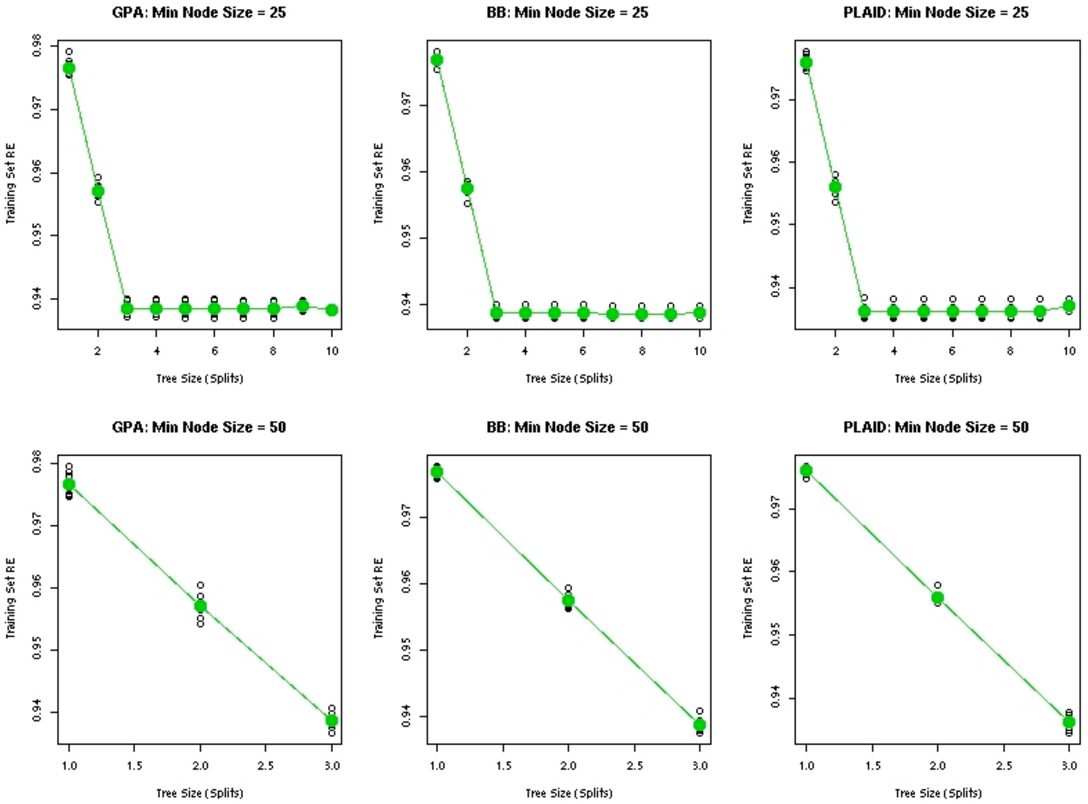
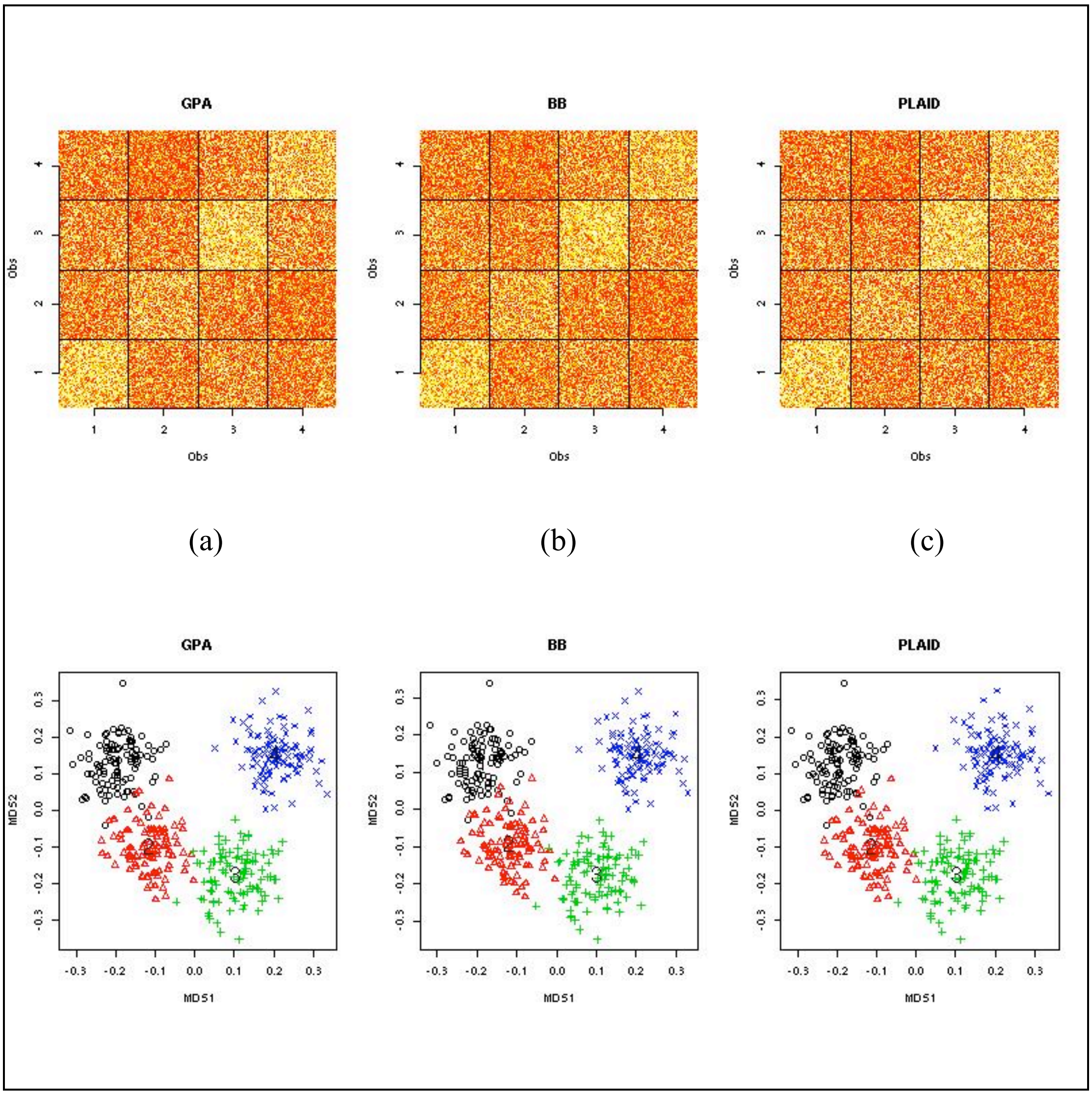

GPA: Min Node Size = 25
BB: Min Node Size = 25
PLAID: Min Node Size = 25
GPA: Min Node Size = 50
BB: Min Node Size = 50
PLAID: Min Node Size = 50
Training Set RE
Tree Size (Splits)
0.94
0.95
0.96
0.97
0.98
2
4
6
8
10
1.0
1.5
2.0
2.5
3.0

Figure 36: Four group simulation consensus configurations.



(a) (b) (c)

Table 4: Four group simulation MCT misclassification performance (Min node size = 25)

	GPA	BB	PLAID
SS	(0,0,0,0) **0 %**	(0,0,0,0) **0 %**	(0,0,0,0) **0 %**
MR	(0,0,0,3) **0.0075%**	(0,0,0,0) **0 %**	(0,0,0,3) **0.0075%**
OR	(0,0,0,0) **0 %**	(0,0,0,0) **0 %**	(0,0,0,0) **0 %**
MR-SSR	(0,0,0,0) **0 %**	(0,0,0,0) **0 %**	(0,0,0,0) **0 %**
OR-SSR	(0,0,0,0) **0 %**	(0,0,0,0) **0 %**	(0,0,0,0) **0 %**

Table 5: Four group simulation MCT misclassification performance (Min node size = 50).

	GPA	BB	PLAID
SS	(0,0,0,0) **0 %**	(0,0,0,0) **0 %**	(0,0,0,0) **0 %**
MR	(0,0,0,3) **0.0075%**	(0,0,0,0) **0 %**	(0,0,0,3) **0.0075%**
OR	(0,0,0,0) **0 %**	(0,0,0,0) **0 %**	(0,0,0,0) **0 %**
MR-SSR	(0,0,0,0) **0 %**	(0,0,0,0) **0 %**	(0,0,0,0) **0 %**
OR-SSR	(0,0,0,0) **0 %**	(0,0,0,0) **0 %**	(0,0,0,0) **0 %**

Table 6: Four group simulation comparative method results.

	GPA	BB	PLAID
AGNES (average)	(2,1,1,0) **0.01 %**	(2,2,5,0) **0.0225 %**	(4,4,9,2) **0.0475 %**
ANGES (complete)	(2,0,3,1) **0.015 %**	(4,1,7,12) **0.06%**	(16,2,26,2) **0.115 %**
ANGES (ward)	(2,0,3,1) **0.015 %**	(4,1,7,12) **0.06%**	(16,2,26,2) **0.115 %**
PAM	(26,4,20,9) **0.14475 %**	(26,5,24,8) **0.1575 %**	(35,27,30,11) **0.2575 %**
K-Means	(0,0,0,0) **0 %**	(0,0,0,0) **0 %**	(0,0,0,0) **0 %**

Growing all MCTs to 3 splits or 4 groups, the classification performances of MCTs are assessed. The results of the misclassifications (Table 4, Table 5) show that the MR method is the only method that misclassifies. These performances were for the PLAID and GPA combination methods, with a terminal node size of 25, where 3 observations in group 4 were mislabelled. On comparison with the standard techniques also run to find 4 groups (Table 6) MCTs performed on par with K-Means and outperformed all other techniques.

6.1.3 Simulation Test 2: Ten uneven but clear groups

The goal of this analysis is to test the sensitivity of MCTs, particularly the splitting methods, to uneven groups. To do this, ten clearly separable but unevenly sized groups are simulated. The group sizes are quite diverse with the smallest being 20 observations and the largest being 180 and the total number of observations is 700. The experiment parameters are:

- The random seed is initially set at 125
- The group size is {75,25,80,125,25,20,100,20,50,180}.
- The maximum count for any element in the RFP is 100.
- The original group configuration is in Table 7.
- Six blurred structures are generated, by adding and subtracting uniform random numbers between 0 and {10,20,30,40,50,60} to the original RFP. In the MDS plots of these RFPs (Figure 38) this is denoted by a +/- within the title.

Table 7: Ten group simulation base configuration group centroids.

		Group									
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Group	**1**	**93**	10	27	32	87	88	48	30	59	55
	2	10	**98**	69	32	9	50	38	38	84	61
	3	27	69	**70**	34	66	62	26	52	5	69
	4	32	32	34	**96**	15	0	68	5	86	42
	5	87	9	66	15	**93**	54	5	7	80	38
	6	88	50	62	0	54	**95**	26	79	24	9
	7	48	38	26	68	5	26	**95**	68	50	27
	8	30	38	52	5	7	79	68	**100**	58	62
	9	59	84	5	86	80	24	50	58	**95**	43
	10	55	61	69	42	38	9	27	62	43	**75**

The original configuration (Figure 37) shows ten quite clear groups. The large groups (10 (180 obs), 4 (125 obs) and 7 (100 obs)) appear at the corners with the smaller groups surrounding them. Over the course of the blurring (Figure 38), the smaller groups merge in with the larger. Groups 3, 2 and 8 blur with 10, 4 and 1 blur with groups 7, and groups 5 and 6 blur together.

Figure 37: Ten group simulation original configuration.

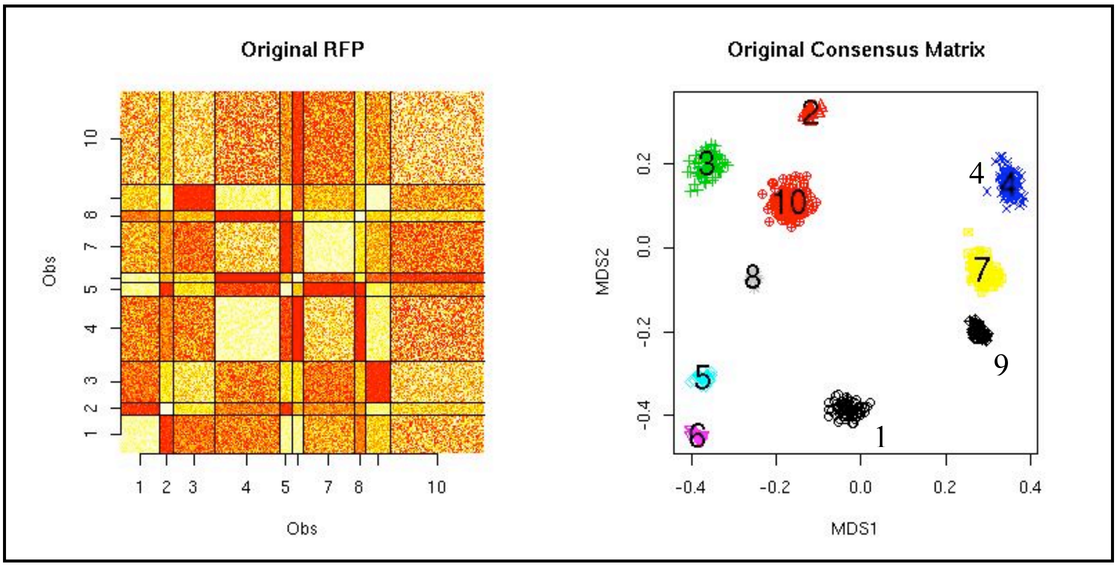
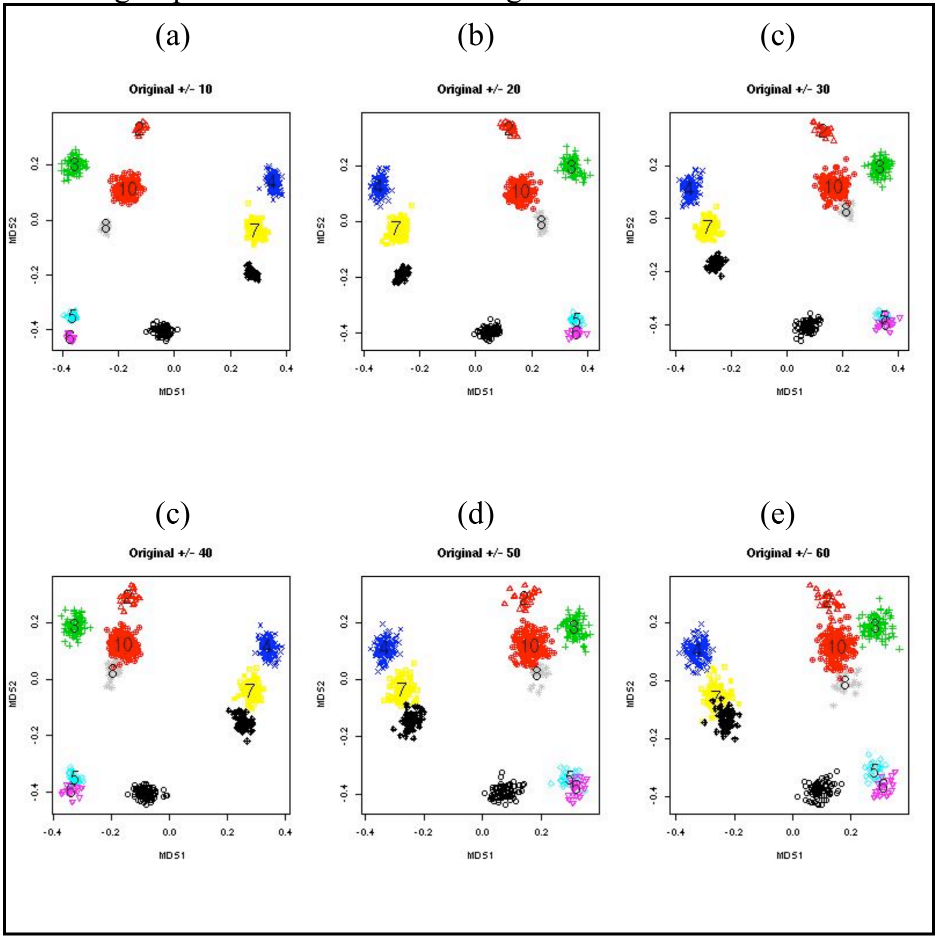


Figure 38: Ten group simulation blurred configurations.



The consensus generation methods are now run on the blurred configurations in (Figure 38). Each of the combination techniques perform about the same in terms of RMSE (Figure 39), each showing an error of between 15 % and 18 % with the base configuration. PLAID combination is clearly doing the worst, with a misclassification rate over twice that of either BB or GPA. However by a side-by-side comparison of the consensus configurations (Figure 40), the combination methods are inseparable.

A raw sum of the residuals ,

$$\text{Residual Sum} = \sum_{i=1}^{N}\sum_{j=1}^{N}\left(c_{ij}^{\text{Orignal}} - \hat{c}_{ij}\right) \tag{6.1}$$

where c_{ij}^{Original} is a count within the base configuration and $\hat{c}_{ij}$ a count within the consensus matrix shows the bias between the consensus matrices and the original. The result of this (Table 8) reveals that the PLAID combination method consistently underestimates the base configuration, as the residual sum is positive. These results imply that PLAID combination is finding the correct structure but at a reduced accuracy.

Figure 39: Ten group simulation consensus RMSEs.

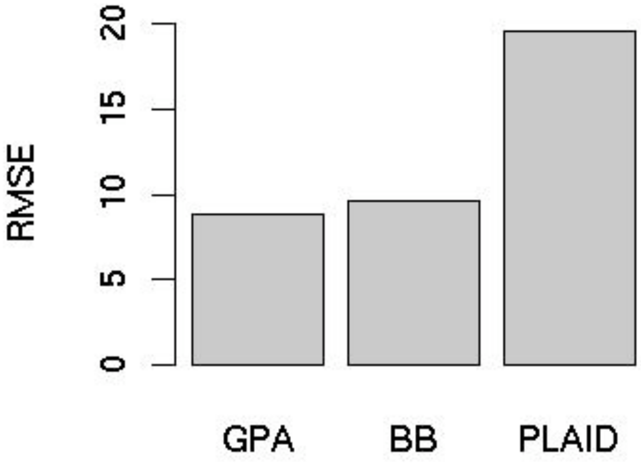


Figure 40: Ten group simulation consensus configurations.

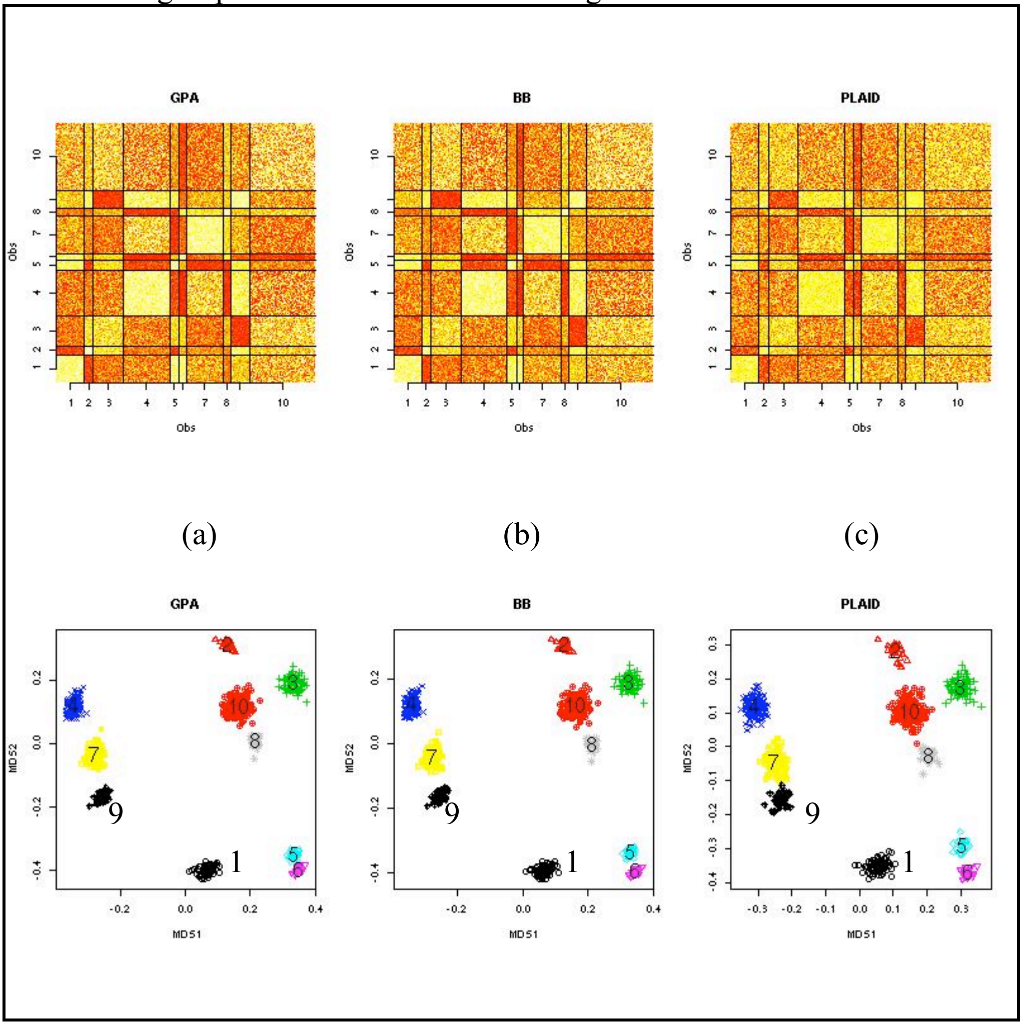
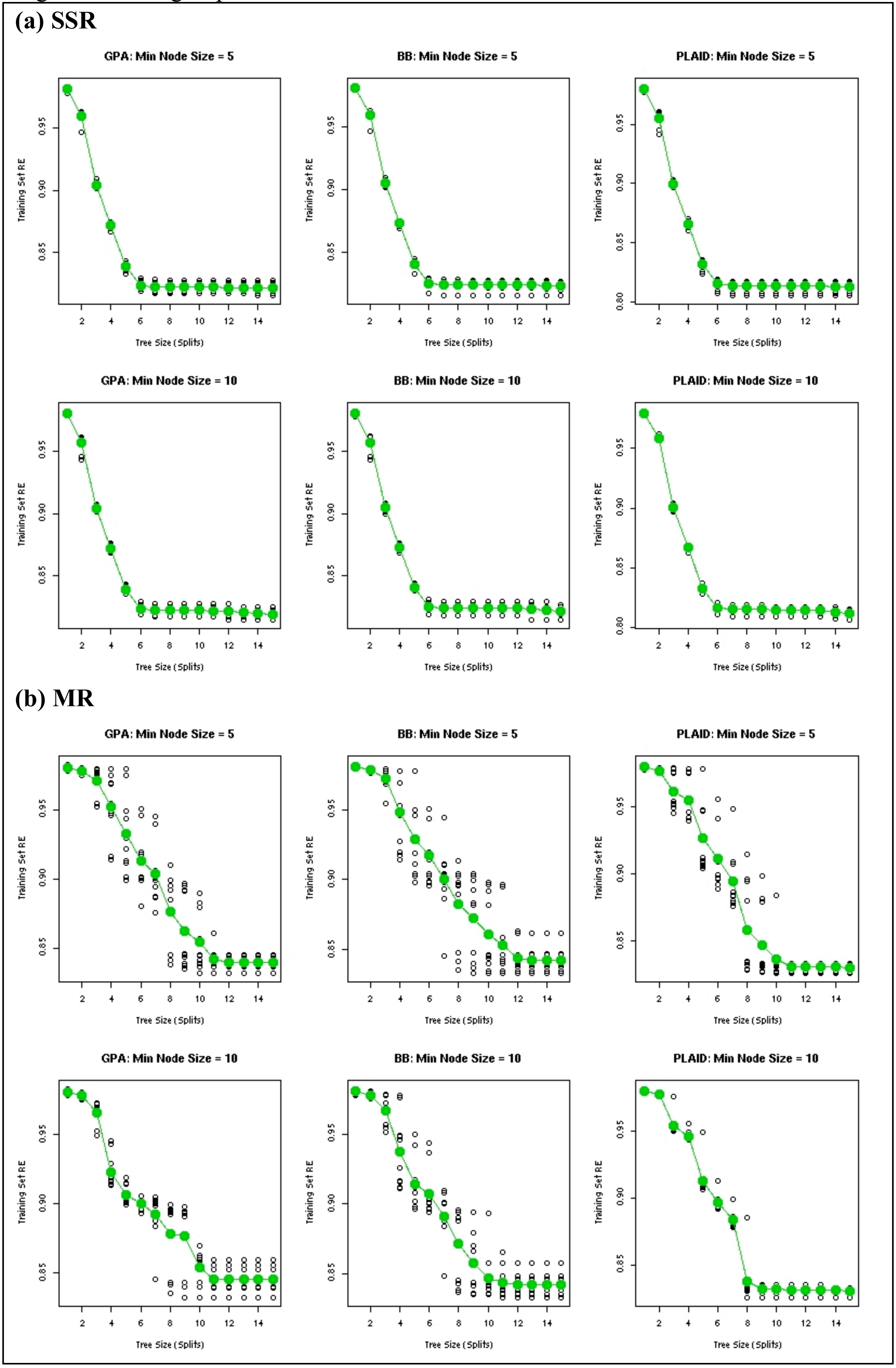

(a) (b) (c)

Table 8: Ten group simulation residual sums.

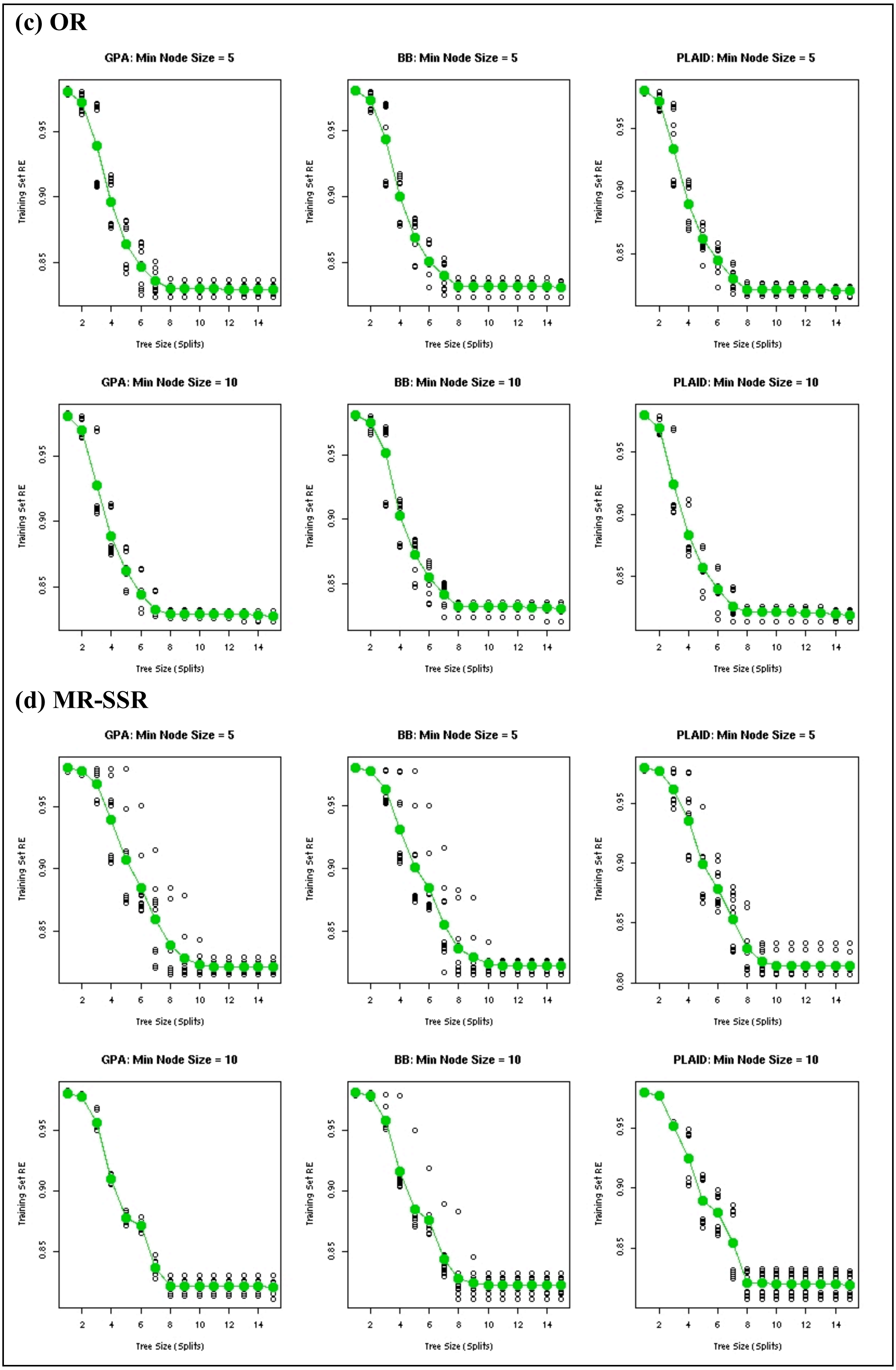
	GPA	BB	PLAID
Residual Sums	29105.05	-156389.8	4290265

Figure 41: Ten group simulation RE curves.

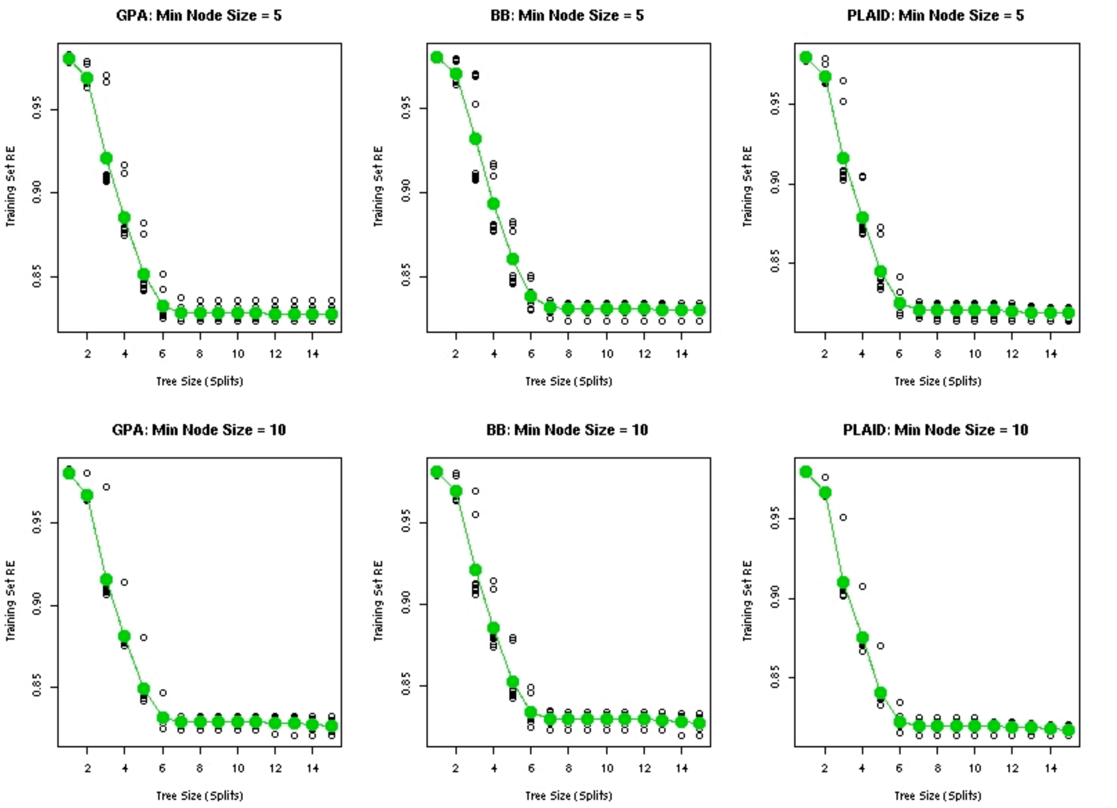


(a) SSR

(b) MR


(c) OR
GPA: Min Node Size = 5
BB: Min Node Size = 5
PLAID: Min Node Size = 5
GPA: Min Node Size = 10
BB: Min Node Size = 10
PLAID: Min Node Size = 10
(d) MR-SSR
GPA: Min Node Size = 5
BB: Min Node Size = 5
PLAID: Min Node Size = 5
GPA: Min Node Size = 10
BB: Min Node Size = 10
PLAID: Min Node Size = 10
Training Set RE
0.80
0.85
0.90
0.95
Tree Size (Splits)
2
4
6
8
10
12
14

(e) OR-SSR



The RE graphs (Figure 41) consistently indicate the presence of 7 to 13 groups depending on the splitting method. The RE curves of SSR, OR and OR-SSR give a clear elbow at 7 or 8 groups (6 or 7 splits), and the MR and MR-SSR elbow placements range from 9 to 13 groups (8 to 12 splits). In comparing the overall structure of the RE curves between splitting methods it is clear that SSR, OR and OR-SSR have an obvious RE structure, and the MR based methods show poor predictions and elbow placements. No real difference is observed between the RE curves for the 5 and 10 minimum terminal node sizes, except with MR splitting on PLAID combining.

The trees are grown to 10 terminal nodes to assess misclassification performance. Between the 5 (Table 9) and 10 (Table 10) minimum terminal node size tables, the only observed difference is with the MR criterion with PLAID combining, where increasing the terminal node size improved performance. It is clear from the misclassification pattern that all methods are completely misclassifying the smaller groups (2 and 8) in favour of the larger groups (4, 7, and 10). It is also clear that the MR alone is not performing as well as the others, however MR-SSR performs on par with other methods. An improvement is also noticed with the combination splitting functions OR-SSR and MR-SSR outperforming the SSR method.

When compared with the other techniques (Table 11) it is obvious that the best performance of MCTs (misclassifying 25 observations) is better than all other clustering techniques, and the worse performances are on par with K-means. Of the standard clustering techniques PAM is performing the best by misclassification of 36 observations on the BB consensus.

Table 9: Ten group simulation MCT misclassification table (Min node size = 5).

	GPA	**BB**	**PLAID**
SSR	(0,25,0,0,0,0,0,20,0,0) **0.064 %**	(0,25,0,0,0,0,0,20,0,0) **0.064 %**	(0,25,0,0,0,0,0,20,0,0) **0.064 %**
MR	(0,0,8,18,25,20,0,20,0,0) **0.13 %**	(0,0,7,18,25,20,0,20,0,0) **0.129 %**	(0,25,67,0,0,0,0,20,0,0) **0.16 %**
OR	(0,25,0,0,0,0,0,20,0,0) **0.064 %**	(0,0,5,0,0,0,0,20,0,0) **0.035 %**	(0,25,0,0,0,0,0,20,0,0) **0.064 %**
MR-SSR	(0,0,6,0,0,0,0,20,0,0) **0.037 %**	(0,0,5,0,0,0,0,20,0,0) **0.035 %**	(0,0,18,0,0,20,0,20,0,0) **0.082 %**
OR-SSR	(0,25,0,0,0,0,0,20,0,0) **0.064 %**	(0,25,0,0,0,0,0,20,0,0) **0.064 %**	(0,25,0,0,0,0,0,20,0,0) **0.064 %**

Table 10: Ten group simulation MCT misclassification table (Min node size = 10).

	GPA	BB	PLAID
SSR	(0,25,0,0,0,0,0,20,0,0) **0.064 %**	(0,25,0,0,0,0,0,20,0,0) **0.064 %**	(0,25,0,0,0,0,0,20,0,0) **0.064 %**
MR	(0,0,8,0,25,20,0,20,0,0) **0.104 %**	(0,0,7,25,20,0,20,0,0) **0.103 %**	(0,0,17,0,0,0,0,20,0,0) **0.053 %**
OR	(0,25,0,0,0,0,0,20,0,0) **0.064 %**	(0,0,5,0,0,0,0,20,0,0) **0.035 %**	(0,25,0,0,0,0,0,20,0,0) **0.064 %**
MR-SSR	(0,0,6,0,0,0,0,20,0,0) **0.037 %**	(0,0,5,0,0,0,0,20,0,0) **0.035 %**	(0,0,18,0,0,0,0,20,0,0) **0.054 %**
OR-SSR	(0,0,6,0,0,0,0,20,0,0) **0.037 %**	(0,25,0,0,0,0,0,20,0,0) **0.064 %**	(0,25,0,0,0,0,0,20,0,0) **0.064 %**

Table 11: Ten group simulation comparative method results.

	GPA	BB	PLAID
AGNES (average)	(0,25,80,0,25,20,100,20,50,0) **0.457 %**	(0,25,80,0,25,20,100,20,50,0) **0.457 %**	(0,25,80,0,25,20,100,20,50,0) **0.457 %**
ANGES (complete)	(0,25,0,0,25,20,100,20,50,19) **0.37 %**	(0,25,0,0,25,20,100,20,50,0) **0.34 %**	(0,25,0,0,25,20,100,20,50,22) **0.374 %**
ANGES (ward)	(0,25,0,0,25,20,100,20,50,19) **0.37 %**	(0,25,0,0,25,20,100,20,50,0) **0.34 %**	(0,25,0,0,25,20,100,20,50,22) **0.374 %**
PAM	(0,25,0,0,0,0,0,0,0,26) **0.073 %**	(0,25,0,0,0,0,0,0,0,11) **0.051 %**	(0,25,0,0,0,0,0,0,0,24) **0.07 %**
K-Means	(0,25,0,0,0,20,0,20,0,0) **0.093 %**	(0,0,0,0,25,20,0,20,0,0) **0.093 %**	(0,25,0,0,25,20,0,20,50,0) **0.20 %**

6.1.4 Simulation Test 3: Addition of pure randomness

The aim of this study is to test the sensitivity of MCTs to the addition of pure random structure. This experiment varies from the others in that the original configuration is entered into RFPs for the combination and tree growing. The key difference is that all other RFPs in the list have no structure and are simply uniform random numbers between 0 and the maximum count. Three clear, equal sized groups are simulated and combined with six purely random configurations. The simulation parameters are:

- The random seed is set initially at 1234567.
- The group size is 50 observations.
- The maximum count for any observation is 50.
- Six purely uniform random configurations are generated with counts ranging from 0 to 50.
- The original configuration is in Table 12.

The original configuration (Figure 42) shows three clear groups, and the random groups (Figure 43) have no obvious grouping structure present. The consensus generation methods are now run with the 6 random configurations and with the original RFP.

Table 12: Pure randomness simulation base configuration group centroids.

		Group		
		1	2	3
Group	1	45	20	5
	2	20	45	25
	3	5	25	45

Figure 42: Pure randomness simulation original configuration.

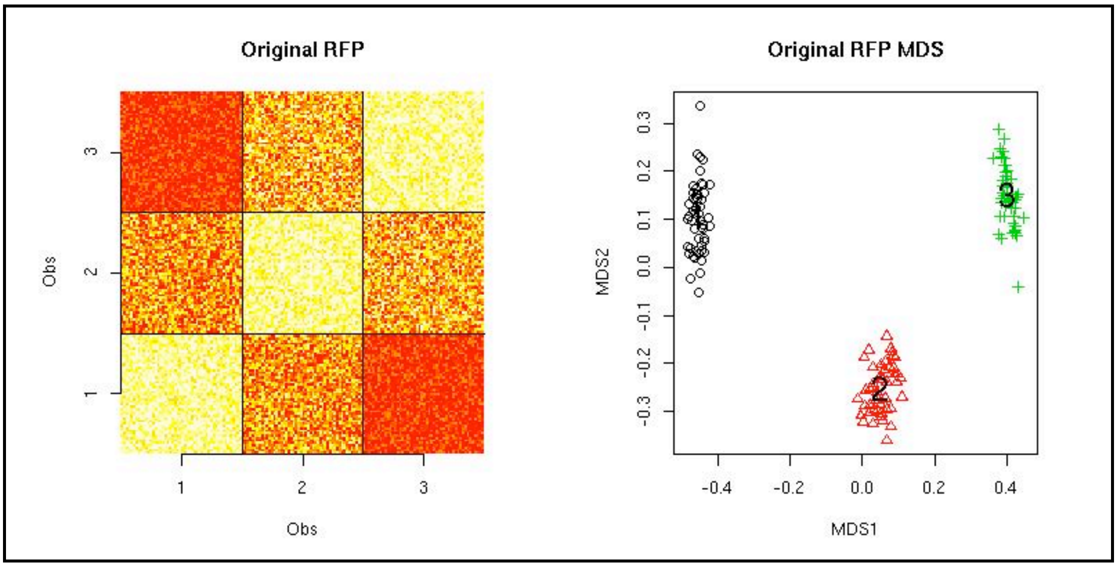


Figure 43: Pure randomness simulation combination method RMSE.

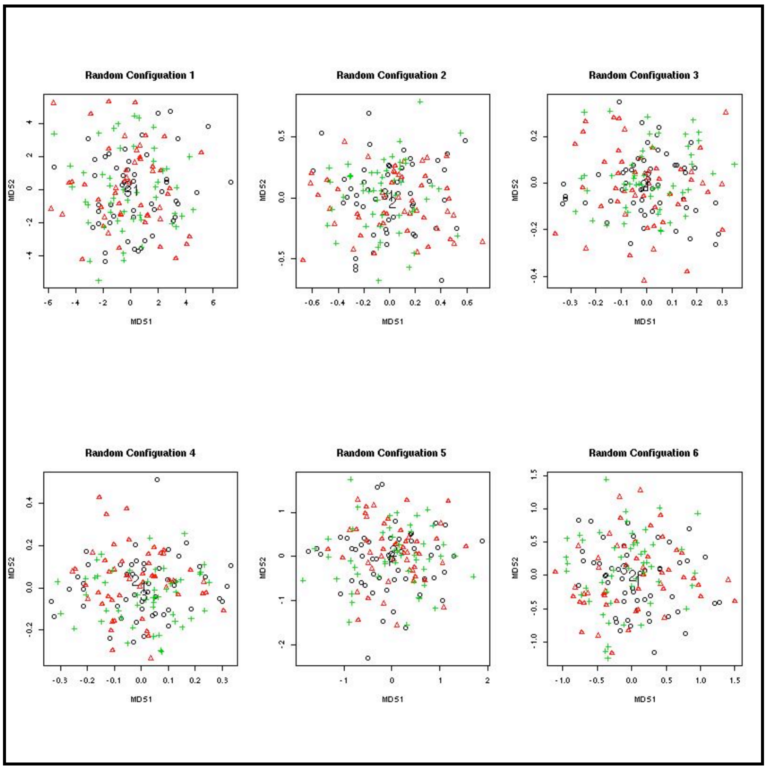
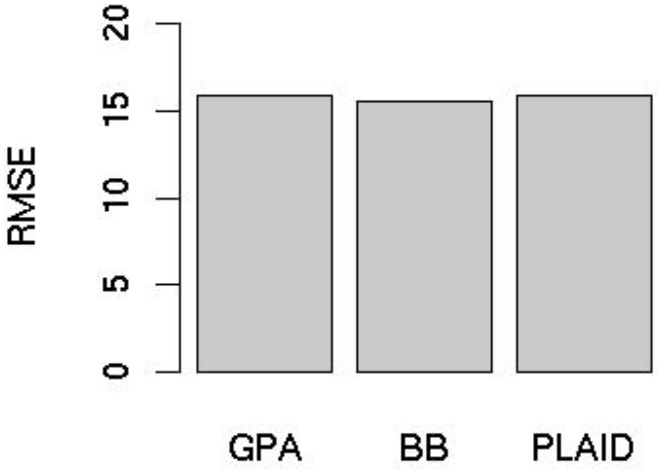
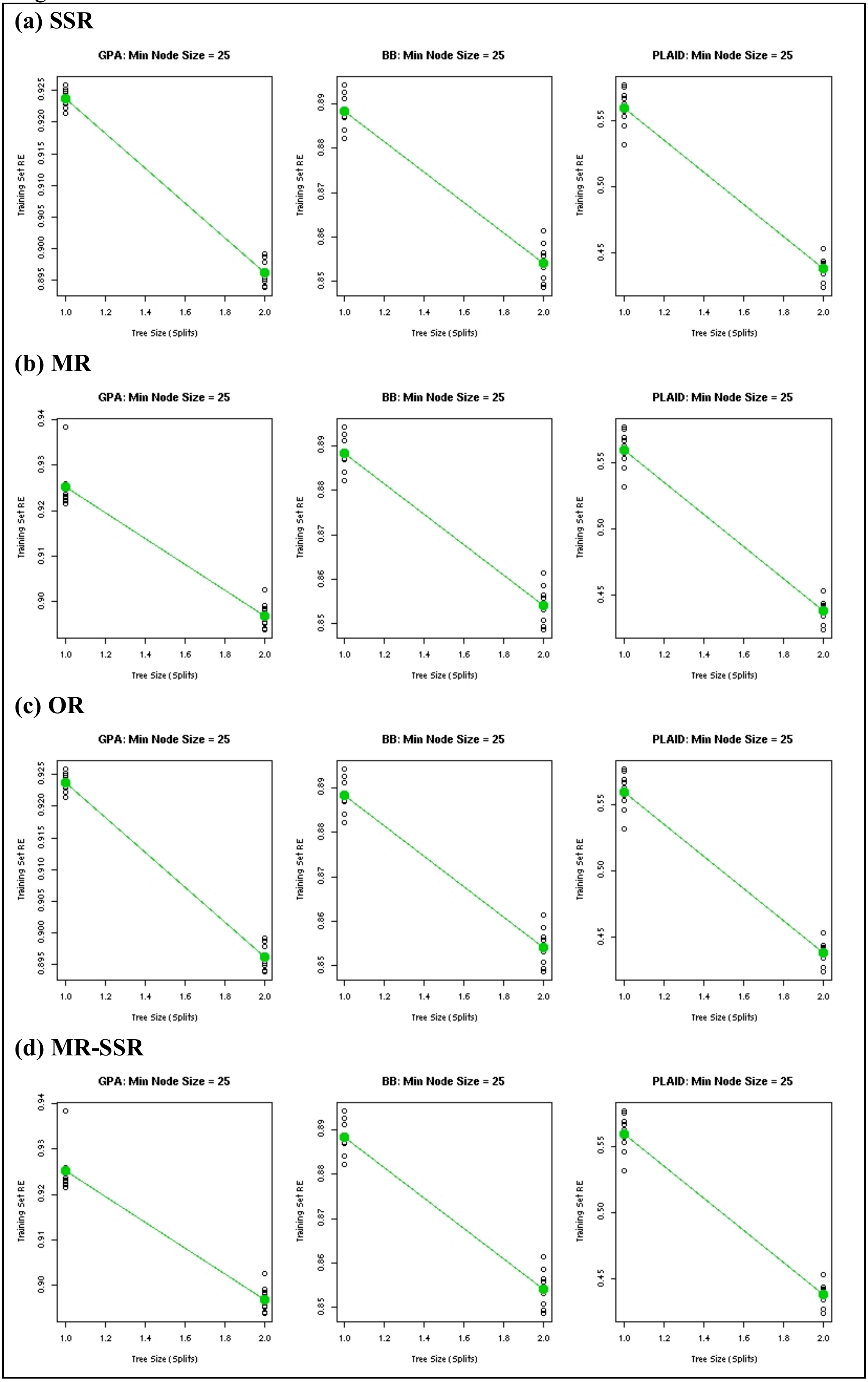


Figure 44: Pure randomness simulation RMSE for the consensus matrices.



From the RMSE results (Figure 43) it is clear that all techniques perform inseparably. The RMSE of approximately 5, indicates the resilience of each technique in the face of randomness. To find the groups a minimum terminal node size was set at 25 observations, half the size of the known groups. It is obvious from the 10-fold cross-validated RE curves (Figure 45) that all splitting methods are consistently finding only three groups (2 splits).

Figure 45: Pure randomness simulation RE curves.



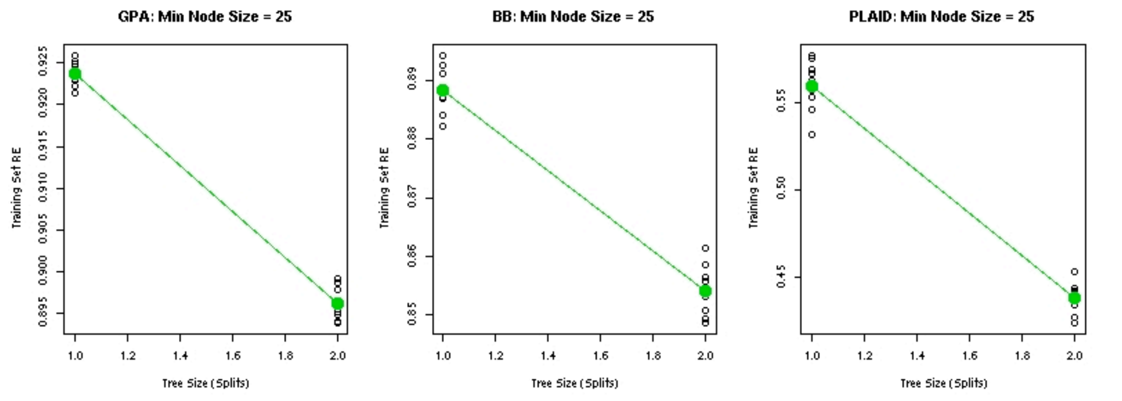
(a) SSR

(b) MR

(c) OR

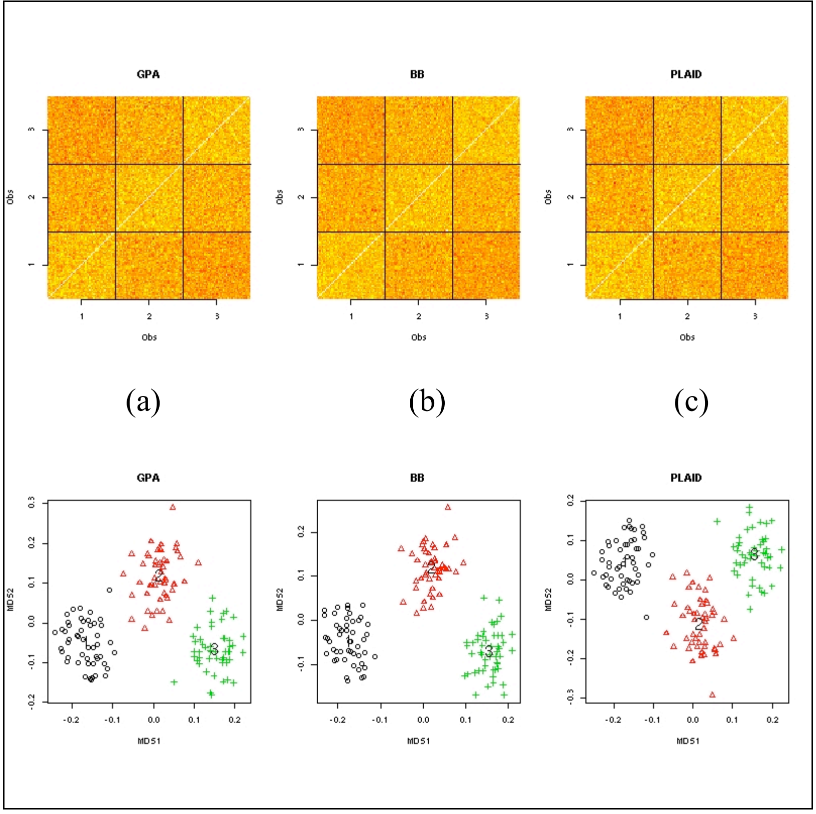
(d) MR-SSR

(e) OR-SSR



A side by side comparison of the consensus matrices (Figure 46) clearly shows the three methods identifying the three group structure. This structure is reinforced with every consensus matrix over all splitting criteria showing no misclassifications (Table 13). On comparison with standard techniques (Table 14) only K-means matches this result. All other techniques mislabel a large percentage of the observations.

Figure 46: Pure randomness simulation consensus configurations.



(a) (b) (c)

Table 13: Pure randomness simulation MCT misclassification results (Min node size = 25).

	GPA	BB	PLAID
SS	(0,0,0) **0 %**	(0,0,0) **0 %**	(0,0,0) **0 %**
MR	(0,0,0) **0 %**	(0,0,0) **0 %**	(0,0,0) **0 %**
OR	(0,0,0) **0 %**	(0,0,0) **0 %**	(0,0,0) **0 %**
MR-SSR	(0,0,0) **0 %**	(0,0,0) **0 %**	(0,0,0) **0 %**
OR-SSR	(0,0,0) **0 %**	(0,0,0) **0 %**	(0,0,0) **0 %**

Table 14: Pure randomness simulation comparative method results.

	GPA	BB	PLAID
AGNES (average)	(1,3,0) **0.0267%**	(0,49,0) **0.3267 %**	(0,48,0) **0.32 %**
ANGES (complete)	(2,8,0) **0.067 %**	(0,0,5) **0.033%**	(1,5,0) **0.04 %**
ANGES (ward)	(1,2,0) **0.02 %**	(0,1,0) **0.0067 %**	(1,2,0) **0.02 %**
PAM	(3,14,14) **0.21 %**	(2,7,8) **0.113 %**	(3,15,10) **0.1867 %**
K-Means	(0,0,0) **0 %**	(0,0,0) **0 %**	(0,0,0) **0 %**

MCTs have shown to be quite resilient to the addition of pure randomness. In this experiment the groups were still obvious in the MDS plot despite a 6:1 ratio of random noise to group structure. MCTs resolve the 3 groups perfectly, where as the comparative methods show some degree of error.

6.2 Random Forest Sensitivity Analysis Using Vietnam Data

This analysis is used to discuss stability of the underlying random forests required to build the MCT RFPs, and their sensitivity to changes in parameter values. An MCT relies heavily on the stable performance of the random forest. However the performance and stability of random forests relies heavily on tree size and minimum terminal node size. The aim of this experiment is to quantify the effect of the relationship between the two parameters, using an applied example where not all the known groups are obvious.

The Vietnam dataset (Table 15) (Hong 1997) contains 18 variables on 224 observations, comprising of 17 continuous profiling variables and one labelling the 6 groups present within the dataset. The variables relate to mineral and heavy metal concentrations within the hair of two cohorts of the Vietnamese population, the first group is regularly exposed to coal, and the second is rarely exposed to coal.

The random forest parameter values that will be varied are:

1. RF tree sizes of {1, 3, 5, 8, 10}.
2. RF minimum terminal node sizes of {5, 10, 15}.

The MCT split method; tree size and number of trees within each random forest are kept constant over all analyses. To determine an appropriate splitting method a pre-analysis with reasonable parameters values (RF tree size 10, and minimum terminal node size of 15) is run. Based on the RE curves (Figure 47) the MR-SSR splitting criterion was selected. The MCT tree size is run to 5 splits, as it is known that 6 groups exist in the data. The number of trees within the forest is set at 200 over all analyses. Before any forest is generated the random seed is set at 1234 to ensure reproducibility of the results and fairness across all methods. This analysis will be performed for both local and global MCTs.

The simulation tests for global MCTs (Table 16) show a clear relationship between tree number and minimum terminal node size. It is known that the smallest group within the data has 18 observations. From the misclassification results it is obvious that for a smaller minimum terminal sizes (5 or 10), to achieve the stable results the size of the trees within the random forest must increase. More so the simulations show that if the trees are not grown such that the number of terminal nodes is at least the number of groups within the dataset, achieving optimal results is not possible.

If the parameters for global MCTs are entered naively, the simulation tests show that quite serious misclassifications can result (Table 16). It seems that choice of combination method does not seriously affect the results. With the exception of PLAID combining with extremely naive parameter values, the resulting misclassification tables are quite similar. The optimal performance of global MCTs for the Vietnam dataset is 17 misclassifications or 7.6 % miss-classification (Figure 48).

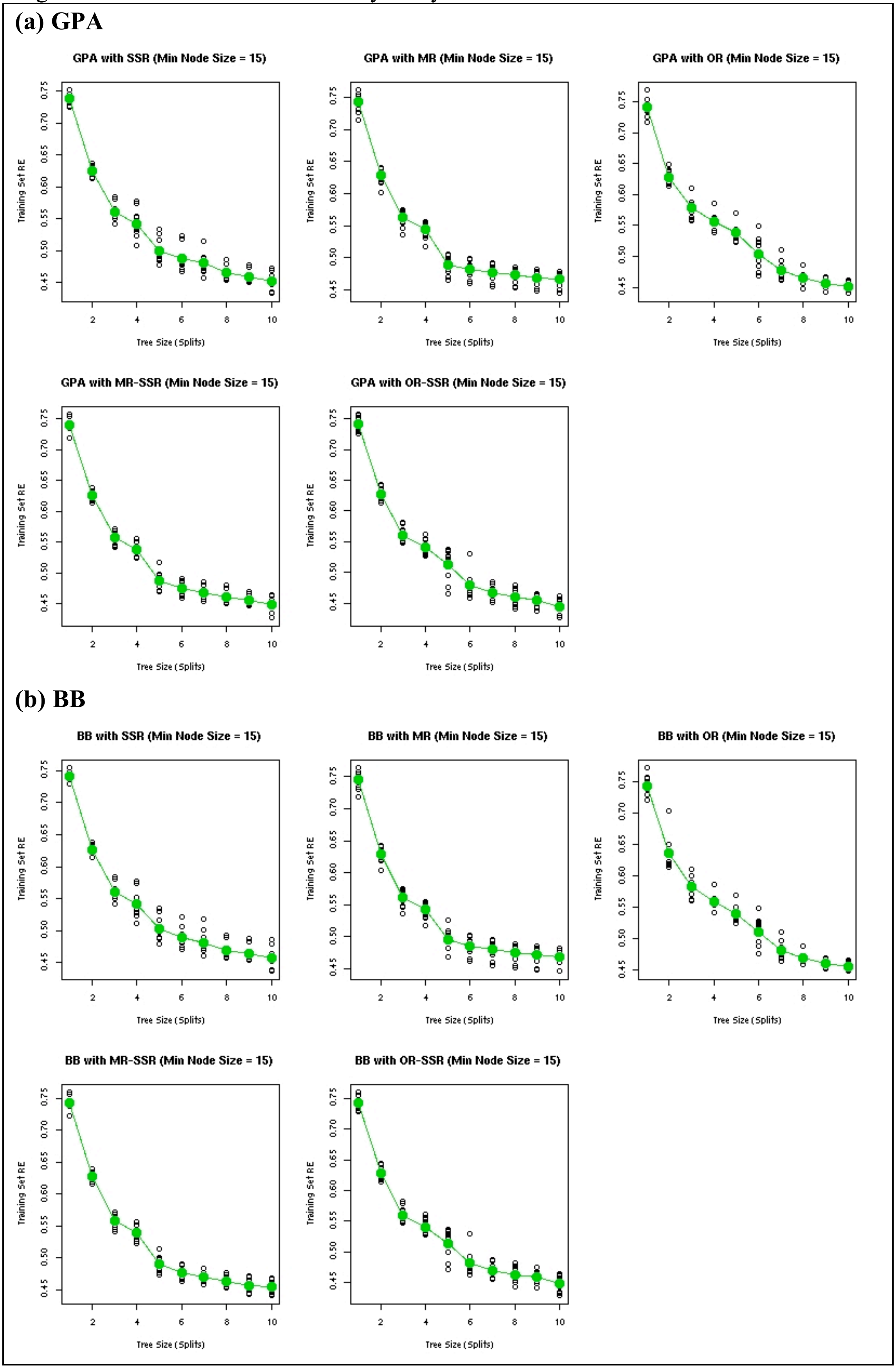
For local MCTs, when compared to global MCTs, the simulations (Table 17) show a stronger dependence on minimum terminal node size. Optimal results were only found at terminal node size of 15, and converged after the tree size was grown to 3 or more splits. A secondary effect for a local MCT is found to be the choice of combination method, with PLAID combining performing worst, inducing 6 more misclassifications when compared to BB or GPA. Overall the optimal local MCT (Figure 49) misclassifies 25 observations or 11.16 % misclassification.

When comparing the splits and the MDS plots between local and global MCTs the initial obvious difference is structure within the MDS plots. Local MCTs show a stronger grouping structure within the MDS plots. These groups correspond to the first three splits within the tree. These splits are identical to those found by the global MCT. The differences between the two trees only occur when distinguishing the last nodes, 12, 13, 14 and 15. This indicates the presence of four strong groups defined by the first three splits.

Table 15: Random forest sensitivity analysis Vietnam dataset description.

Variable	Description	Type
zlas	*Standardised logarithm of arsenic concentration*	Continuous
zlba	*Standardised logarithm of barium concentration*	Continuous
zlcd	*Standardised logarithm of cadmium concentration*	Continuous
zlcr	*Standardised logarithm of chromium concentration*	Continuous
zlcu	*Standardised logarithm of copper concentration*	Continuous
zlhg	*Standardised logarithm of mercury concentration*	Continuous
zlmn	*Standardised logarithm of manganese concentration*	Continuous
zlmo	*Standardised logarithm of molybdenum concentration*	Continuous
zlni	*Standardised logarithm of nickel concentration*	Continuous
zlpb	*Standardised logarithm of lead concentration*	Continuous
zlse	*Standardised logarithm of selenium concentration*	Continuous
zlsn	*Standardised logarithm of tin concentration*	Continuous
zlsr	*Standardised logarithm of strontium concentration*	Continuous
zlth	*Standardised logarithm of thorium concentration*	Continuous
zlti	*Standardised logarithm of titanium concentration*	Continuous
zlu	*Standardised logarithm of uranium concentration*	Continuous
zlv	*Standardised logarithm of vanadium concentration*	Continuous
grp	*Known group*	Nominal (1) **Control Adults**: Adults with low exposure to coal. (*n=31*) (2) **Miner Adults**: Males employed at a coal mine. (*n=56*) (3) **Burner Adults**: Females using coal for cooking. (*n=18*) (4) **Control Children**: Children with low exposure to coal. (*n=31*) (5) **Miner Children**: Children of coal miners. (*n=47*) (6) **Burner Children**: Children with exposure to coal through its use for cooking. (i)

Figure 47: Random forest sensitivity analysis RE curves.



(a) GPA

(b) BB

(c) PLAID

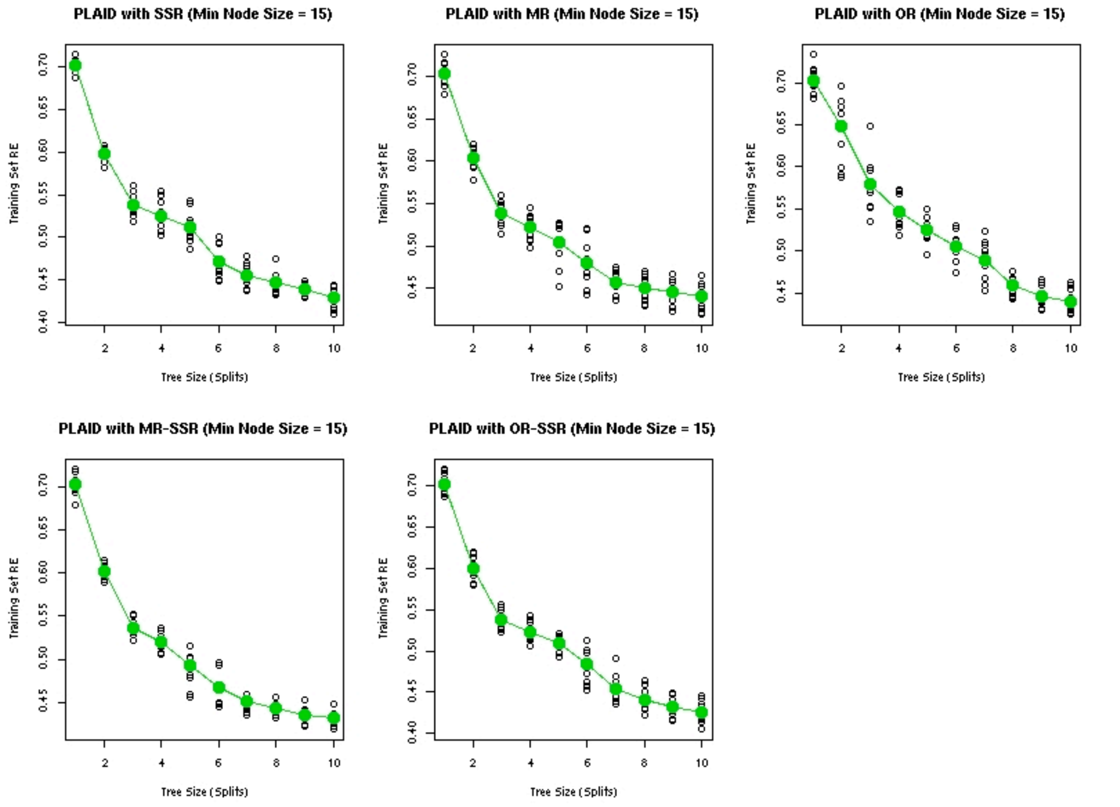

PLAID with SSR (Min Node Size = 15)
PLAID with MR (Min Node Size = 15)
PLAID with OR (Min Node Size = 15)
PLAID with MR-SSR (Min Node Size = 15)
PLAID with OR-SSR (Min Node Size = 15)
Training Set RE
Tree Size (Splits)
0.40
0.45
0.50
0.55
0.60
0.65
0.70
2
4
6
8
10

Table 16: Random forest sensitivity analysis global MCT misclassification tables. The number of observations misclassified for each group is presented in brackets. The optimal performance for each combination method is emboldened.

(a) GPA

		Minimum Terminal Node Size		
		5	**10**	**15**
Maximum tree size	**1**	(1,0,18,1,8,13) 0.18 %	(31,2,18,31,6,17) 0.47 %	(31,2,18,31,6,14) 0.46 %
	3	(1,0,18,1,8,13) 0.183 %	(1,0,18,1,8,13) 0.183 %	(1,0,3,1,3,13) 0.094 %
	5	(1,2,18,4,6,12) 0.19 %	(1,0,3,1,3,13) 0.094 %	**(1,0,3,1,3,9)** **0.076 %**
	8	(1,2,3,4,1,10) 0.094	**(1,0,3,1,3,9)** **0.076 %**	**(1,0,3,1,3,9)** **0.076 %**
	10	**(1,0,3,1,3,9)** **0.076 %**	**(1,0,3,1,3,9)** **0.076 %**	**(1,0,3,1,3,9)** **0.076 %**

(b) BB

		Minimum Terminal Node Size		
		5	**10**	**15**
Maximum tree size	**1**	(1,0,18,1,8,13) 0.183 %	(31,14,18,31,1,20) 0.513 %	(31,14,3,31,1,20) 0.446 %
	3	(1,0,18,1,8,13) 0.183 %	(1,0,18,1,8,13) 0.183 %	(1,0,3,1,3,13) 0.094 %
	5	(1,2,18,4,6,12) 0.19 %	(1,0,3,1,3,13) 0.094 %	**(1,0,3,1,3,9)** **0.076 %**
	8	(1,2,3,4,1,14) 0.11 %	**(1,0,3,1,3,9)** **0.076 %**	**(1,0,3,1,3,9)** **0.076 %**
	10	**(1,0,3,1,3,9)** **0.076 %**	**(1,0,3,1,3,9)** **0.076 %**	**(1,0,3,1,3,9)** **0.076 %**

(c) PLAID

		Minimum Terminal Node Size		
		5	**10**	**15**
Maximum tree size	**1**	(6,0,18,12,44,39) 0.53 %	(27,29,4,7,41) 0.48 %	(31,16,18,31,8,41) 0.65 %
	3	(1,0,18,1,44,41) 0.47 %	(1,0,18,0,8,13) 0.18 %	(1,0,3,1,3,13) 0.094 %
	5	(1,2,18,4,6,12) 0.19 %	(1,0,3,1,3,11) 0.085 %	**(1,0,3,1,3,9)** **0.076 %**
	8	(1,2,3,4,1,10) 0.094 %	**(1,0,3,1,3,9)** **0.076 %**	**(1,0,3,1,3,9)** **0.076 %**
	10	(1,0,18,1,3,9) 0.143 %	**(1,0,3,1,3,9)** **0.076 %**	**(1,0,3,1,3,9)** **0.076 %**

Table 17: Random forest sensitivity analysis local MCT misclassification table. The number of observations misclassified for each group is presented in brackets. The optimal performance for each combination is emboldened.

(a) GPA

		Minimum Terminal Node Size		
		5	**10**	**15**
Maximum tree size	**1**	(0,32,18,1,47,41) 0.62 %	(0,27,18,1,14,6) 0.29 %	(0,27,18,1,14,6) 0.29 %
	3	(1,0,18,1,47,41) 0.48 %	(1,2,5,1,8,13) 0.134 %	**(1,12,1,1,4,6)** **0.11 %**
	5	(1,2,18,4,47,41) 0.50 %	(1,12,18,1,3,41) 0.34 %	**(1,12,1,1,4,6)** **0.11 %**
	8	(1,2,18,4,47,41) 0.50 %	(1,12,18,1,3,41) 0.34 %	**(1,12,1,1,4,6)** **0.11 %**
	10	(1,2,18,1,47,41) 0 49 %	(1,12,18,1,3,41) 0.34 %	**(1,12,1,1,4,6)** **0.11 %**

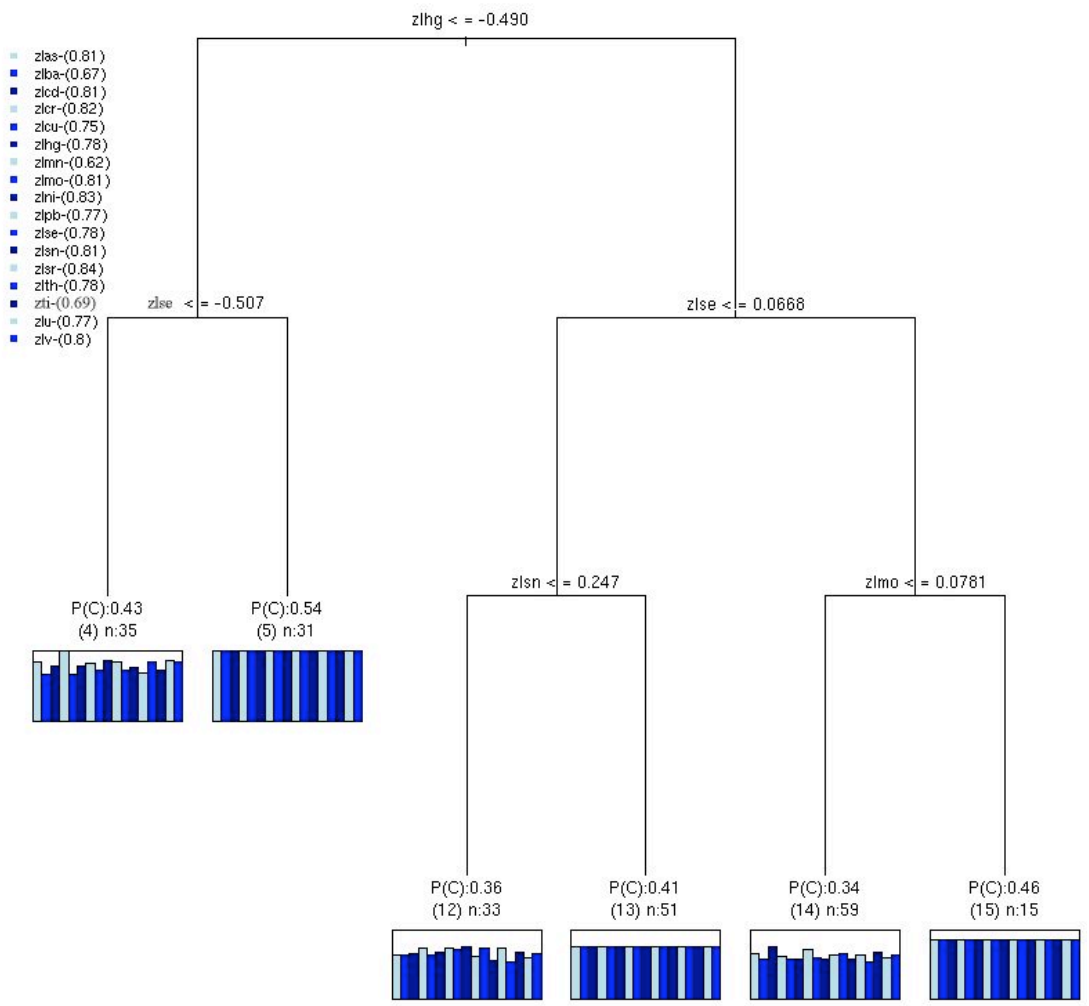
(b) BB

		Minimum Terminal Node Size		
		5	**10**	**15**
Maximum tree size	**1**	(0,23,18,1,0,41) 0.37 %	(0,6,18,1,14,6) 0.20 %	(0,6,18,1,14,6) 0.20 %
	3	(1,0,18,1,4,47,41) 0.50 %	(1,0,5,1,8,13) 0.125 %	**(1,12,1,1,4,6)** **0.11 %**
	5	(1,2,18,4,47,41) 0.504 %	(1,12,1,1,3,41) 0.263 %	**(1,12,1,1,4,6)** **0.11 %**
	8	(1,2,18,4,47,41) 0.504 %	(1,12,1,1,3,41) 0.263 %	**(1,12,1,1,4,6)** **0.11 %**
	10	(1,0,18,1,47,41) 0.48 %	(1,12,1,1,3,41) 0.263 %	**(1,12,1,1,4,6)** **0.11 %**

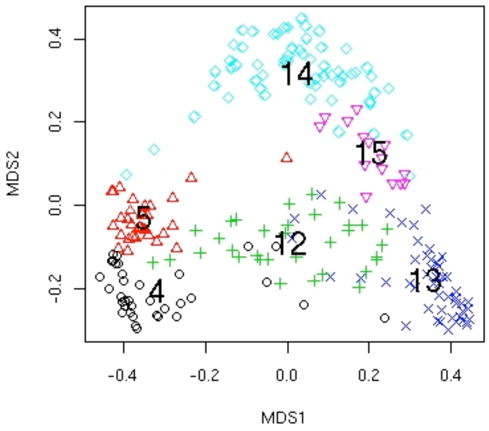
(c) PLAID

		Minimum Terminal Node Size		
		5	**10**	**15**
Maximum tree size	**1**	(1,0,18,1,47,41) 0.48 %	(1,0,6,2,8,13) 0.134 %	(1,0,18,1,8,13) 0.183 %
	3	(1,0,18,1,47,41) 0.48 %	(1,0,5,1,8,13) 0.125 %	(1,0,18,1,8,13) 0.183 %
	5	(1,2,18,4,47,41) 0.504 %	(1,12,18,1,47,41) 0.54 %	**(1,0,18,1,4,6)** **0.134 %**
	8	(1,2,18,4,47,41) 0.504 %	(1,12,18,1,47,41) 0.54 %	**(1,0,18,1,4,6)** **0.134 %**
	10	(1,0,18,1,47,41) 0.48 %	(1,12,18,1,47,41) 0.54 %	**(1,0,18,1,4,6)** **0.134 %**

Figure 48: Random forest sensitivity analysis best global MCT, MR-SSR splitting with GPA consensus, and random forest tree size of 5 with a minimum terminal node size of 15.



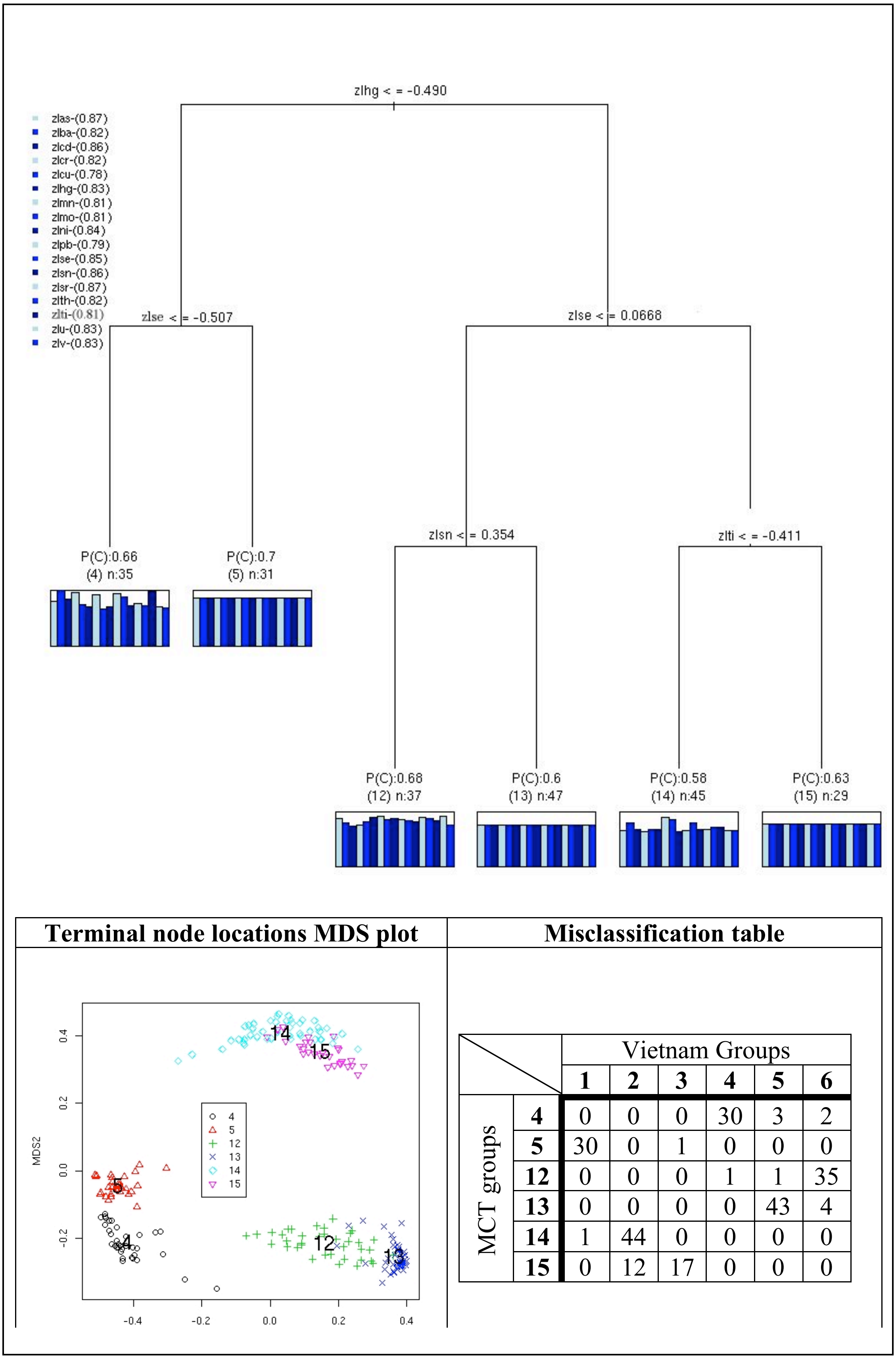
Terminal node locations MDS plot



Misclassification table

		Vietnam Groups					
		1	**2**	**3**	**4**	**5**	**6**
MCT groups	**4**	0	0	0	30	3	2
	5	30	0	1	0	0	0
	12	0	0	0	1	0	32
	13	0	0	0	0	44	7
	14	1	56	2	0	0	0
	15	0	0	15	0	0	0

Figure 49: Random forest sensitivity analysis best local MCT, MR-SSR splitting with GPA consensus, and random forest tree size of 5 with a minimum terminal node size of 15.



Terminal node locations MDS plot

Misclassification table

		Vietnam Groups					
		1	2	3	4	5	6
MCT groups	4	0	0	0	30	3	2
	5	30	0	1	0	0	0
	12	0	0	0	1	1	35
	13	0	0	0	0	43	4
	14	1	44	0	0	0	0
	15	0	12	17	0	0	0

6.3 Summary

MCTs have been shown to be a powerful tool for uncovering groups within a dataset. In three simulation experiments MCTs consistently outperformed PAM and agglomerative techniques and performed comparably to K-Means. Furthermore these experiments show MCTs to be quite resistant to noise within the response set. Even with complete noise variables within the response, MCTs still resolved the correct groups. The RE curves are also highlighted as an accurate tool for estimating the number of groups within a dataset. For simple grouping cases, (simulation tests 1 and 3), these curves estimated the number of groups exactly. As the group structure became more complex the RE curves became less obvious (simulation test 2), however still provided a range of group numbers that encompassed the number of known groups.

The sensitivity analysis on the Vietnam analysis highlights key parameters that affect the construction of an MCT. These parameters are random forest terminal node size and tree size. These parameters have a threshold effect where, if specified correctly MCTs provide a stable optimal performance, otherwise the performances vary considerably. Global MCTs are found to be more stable than local MCTs, and PLAID combining is found to produce the weakest consensus.

7. Benchmark Examples

In this section we present tree-based profiling and clustering methods on three benchmark datasets. Each of these datasets has been selected to highlight features of tree-based methods and to compare their performances. The datasets selected are all freely available benchmark datasets.

The first dataset is the Thyroid dataset, which is a clustering problem involving only quantitative variables. Here the improvement in clustering performance gained through using the auto-association proximity matrices is shown. MCTs are compared with auto-associative random forests and treeboost, AA-MRTs, PAM and K-means.

The second dataset is the Wisconsin breast cancer dataset. This analysis is used to compare the performance of tree methods in a categorical domain with a known clear grouping structure. For this analysis MCT approaches are compared to binary substitution and Gower distance methods.

The third dataset is the horse colic dataset. This analysis is focused on the performance on MCTs in a mixed domain profiling problem. Here the limits of the Gower distance and binary substitution methods are shown and the power of the proximity matrices is highlighted. This study also explores the features of MCTs that assist in further understanding and simplifying the problem. In particular the ability of PLAID consensus generation to find subgroups within variables of the profiling set is highlighted.

7.1 Clustering Quantitative Variables: Thyroid dataset

In this analysis a comparison between tree-based methods for clustering and existing methods is performed using the thyroid dataset. The thyroid dataset (Coomans, Broeckaert, Jonckheer and Massart 1983) consists of 215 observations on 5 variables that describe the action of the thyroid gland. There are three known groups in the data corresponding to hypothyroid (1), hyperthyroid (2) and normal (3) patients. The other variables are hormone levels measured in the blood. These are:

1. TSH
2. DTSH
3. RT3U
4. T4
5. T3

The goal of the analysis is to cluster the data and compare the clustering performance with the known groups.

7.1.1 AA-MRT

The RE graph for AA-MRT is used to determine the size of the tree (Figure 50), and it can be seen that the performance plateaus at 5 terminal nodes, and the corresponding tree is displayed in Figure 51. Using the terminal node locations as group classifications, AA-MRTs misclassify 35 observations (Table 18) when compared with the known groups. The AA-MRT of the raw data outperforms PAM, which misclassified 49 observations, but not K-means, which misclassified 30 observations.

Figure 50: Thyroid analysis AA-MRT RE graph.

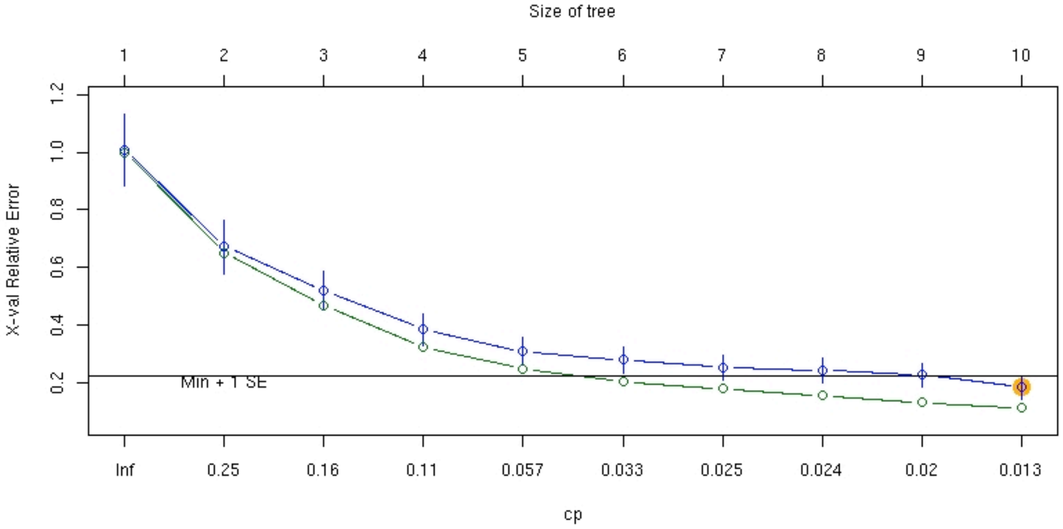


Figure 51: Thyroid analysis AA-MRT.

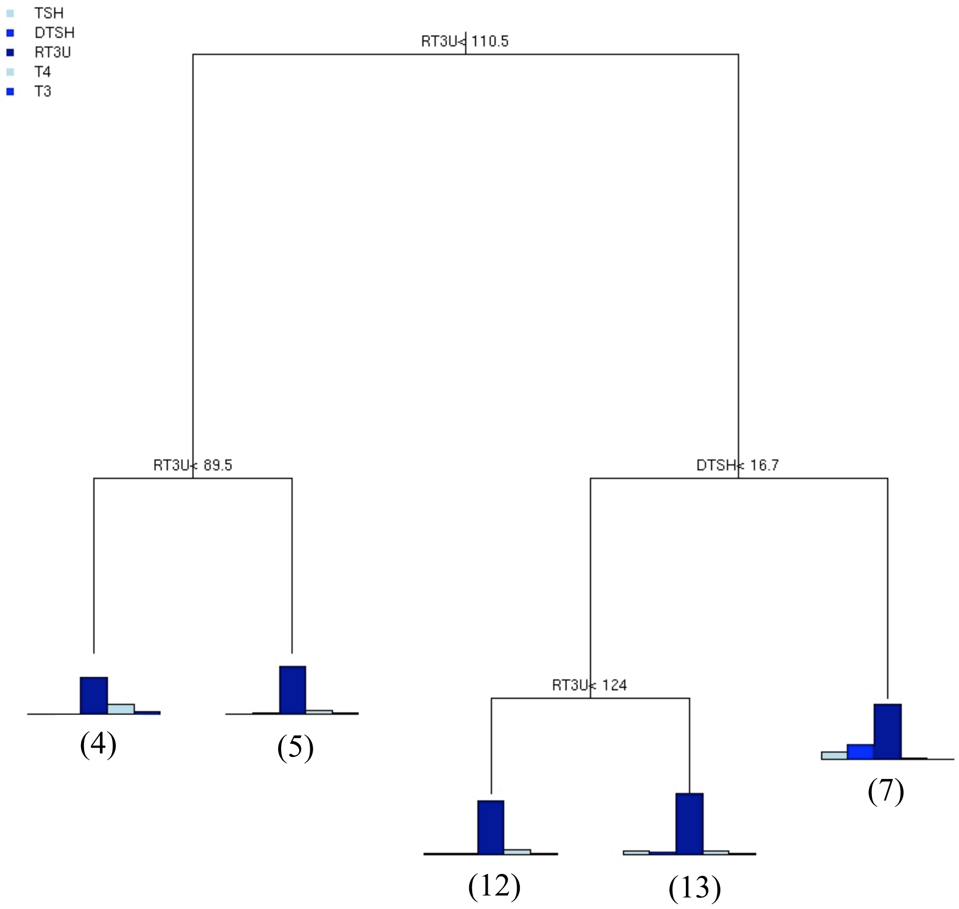


Table 18: Thyroid analysis AA-MRT misclassification table

	Terminal Node				
	4	**5**	**7**	**12**	**13**
Hypothyroid	0	4	4	10	12
Hyperthyroid	16	14	2	3	0
Normal	0	78	64	8	0

7.1.2 AA-RF

AA-RF performs quite well on the dataset, converging to a stable predictive performance after 100 trees are added to the model (Figure 52). This precision is mirrored within the proximity matrix and MDS images with three groups obvious (Figure 53). Clustering this matrix with an MCT using SSR splitting (Figure 54), gives 22 misclassifications (Table 19) and 12 misclassifications are recorded for PAM and 11 for K-means. It is clear that the clustering techniques all do better on the proximities than on the raw data, whereas the fact that K-Means and PAM do better than MCTs is a reflection on the overlapping nature of the groups. If the groups are strongly overlapping it is likely that a partition on a single variable will be sufficient. Both K-means and PAM have the luxury of not requiring a clear single variable separation between the groups and therefore do better.

Figure 52: Thyroid analysis AA-RF error convergence plot.

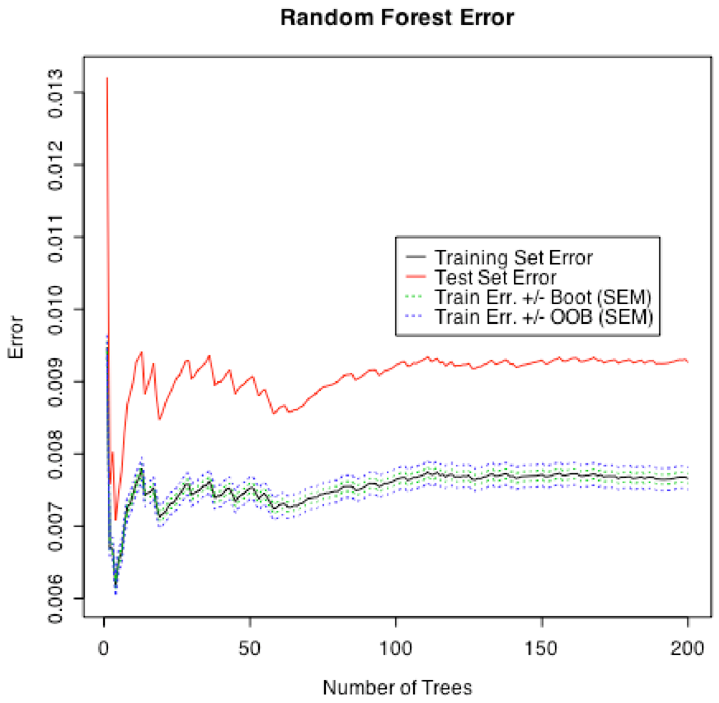


Figure 53: Thyroid analysis AA-RF proximity images.

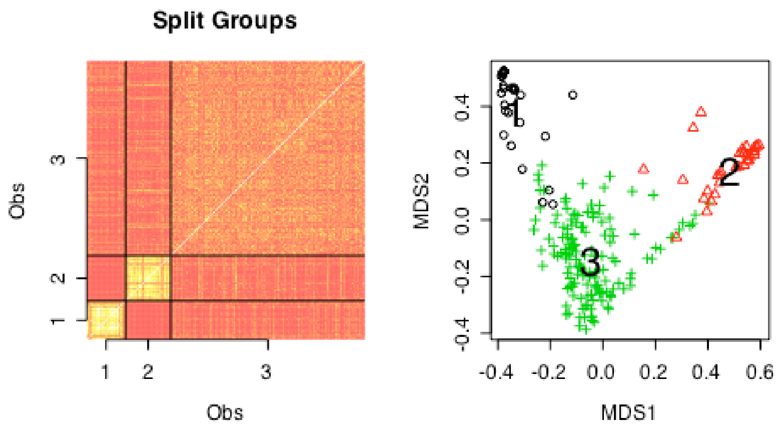


Figure 54: Thyroid analysis SSR partition on the AA-RF proximity matrix.

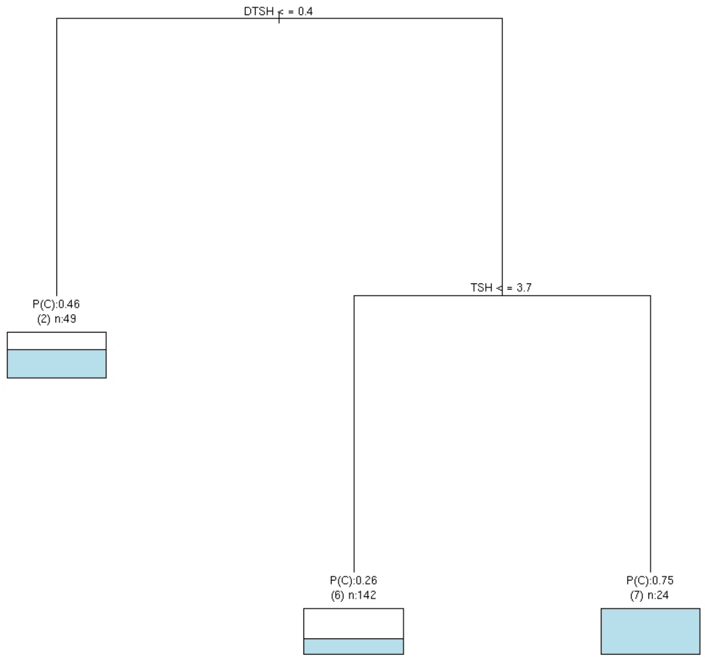


Table 19: Thyroid analysis AA-RF misclassification table

	Terminal Node		
	2	**6**	**7**
Hypothyroid	0	34	15
Hyperthyroid	6	1	135
Normal	24	0	0

7.1.3 AA-Treeboost

The AA-Treeboost model proved to be more complex than the random forest models with the error converging (Figure 55) after 300 trees were added to the model. Despite the number of trees added the proximity images do not obviously show three known groups (Figure 56). This non-obvious structure affects the performances of the base clustering algorithms with K-Means and PAM misclassifying 38 observations. However MCTs with MR splitting (Figure 57) on the treeboost proximity matrix misclassified only 15 observations (Table 20). The improvement gained by MCTs is most likely a direct result of trees being used to construct the proximity matrix, thus allowing MCTs to identify the structure not easily found by other methods.

Figure 55: Thyroid analysis AA-Treeboost error convergence plot.

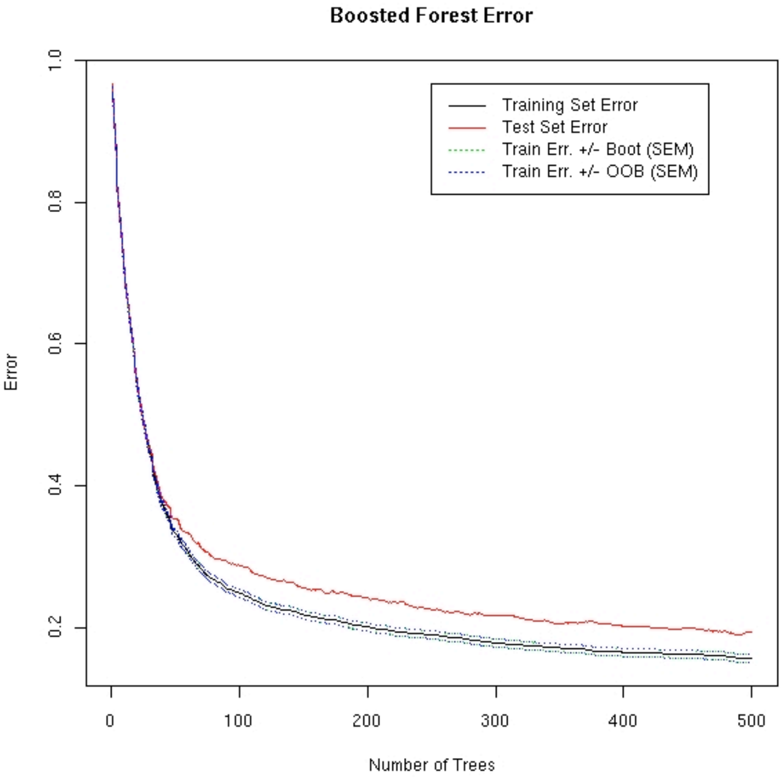


Figure 56: Thyroid analysis AA-Treeboost proximity images.

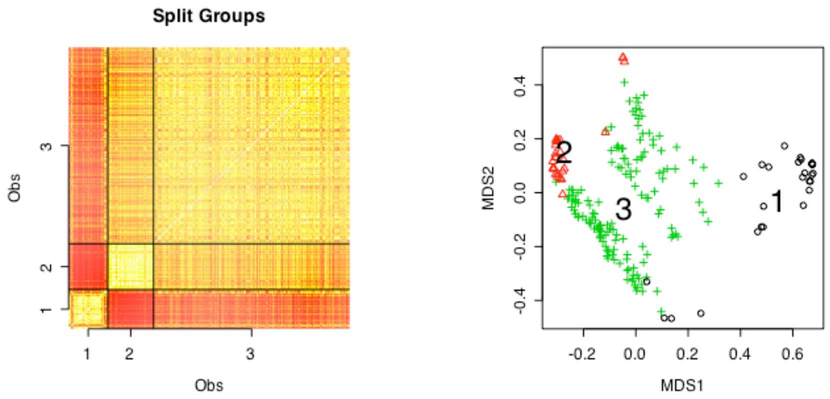


Figure 57: Thyroid analysis MR partition on the AA-Treeboost proximity matrix.

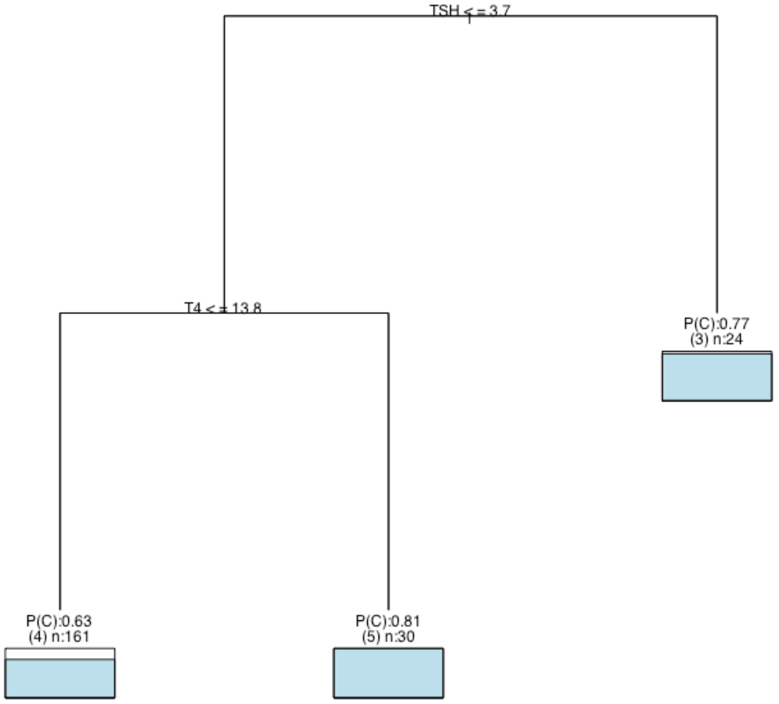


Table 20: Thyroid analysis AA-Treeboost misclassification table.

	Terminal Node		
	3	**4**	**5**
Hypothyroid	24	6	0
Hyperthyroid	0	7	28
Normal	0	148	2

7.1.4 Global MCT

The cross-validation for Global MCTs using SSR splitting identifies 2 splits or three groups (Figure 58). The proximity matrix images (Figure 59) are comparable to those found by AA-RF. The corresponding MCT (Figure 60) misclassifies 19 observations (Table 21). However by observation of each terminal node's probability of expression, "P(C)" it can be seen that node 3, which corresponds to the normal group is under-expressed showing a probability of 0.32. This implies that this group is difficult for trees to correctly classify. A finding that is mirrored by its broad dispersion over the MDS plot (Figure 59). When compared with K-means and PAM, MCTs are found to under-perform as they both only misclassify 9 observations.

Figure 58: Thyroid analysis global MCT 10-Fold cross-validated RE curves.

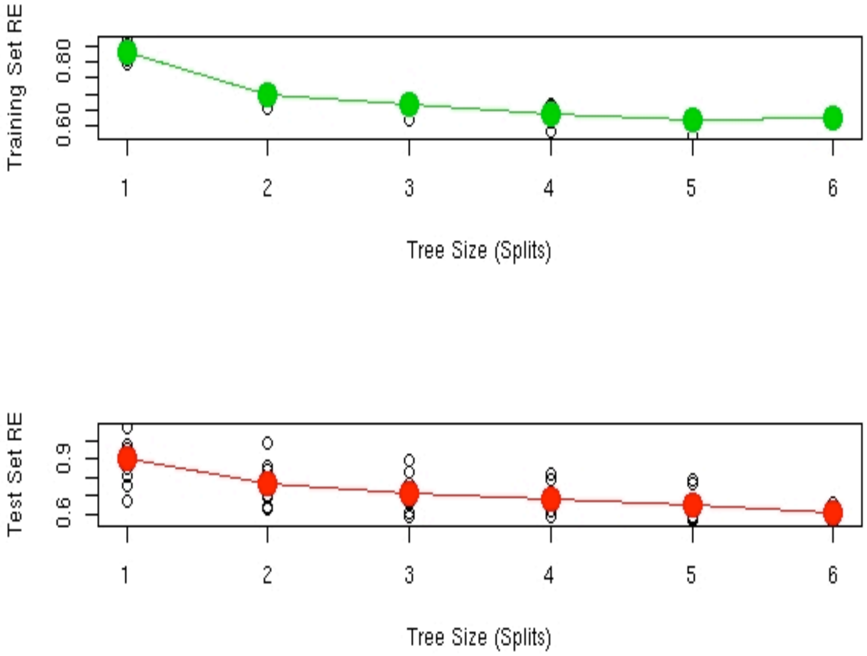


Figure 59: Thyroid analysis global MCT proximity images.

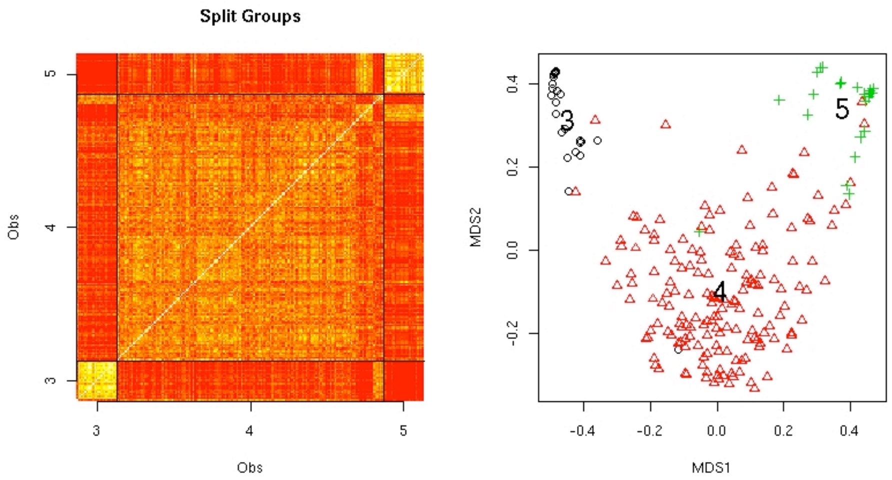


Figure 60: Thyroid analysis global MCT, constructed with MR splitting on the GPA consensus.

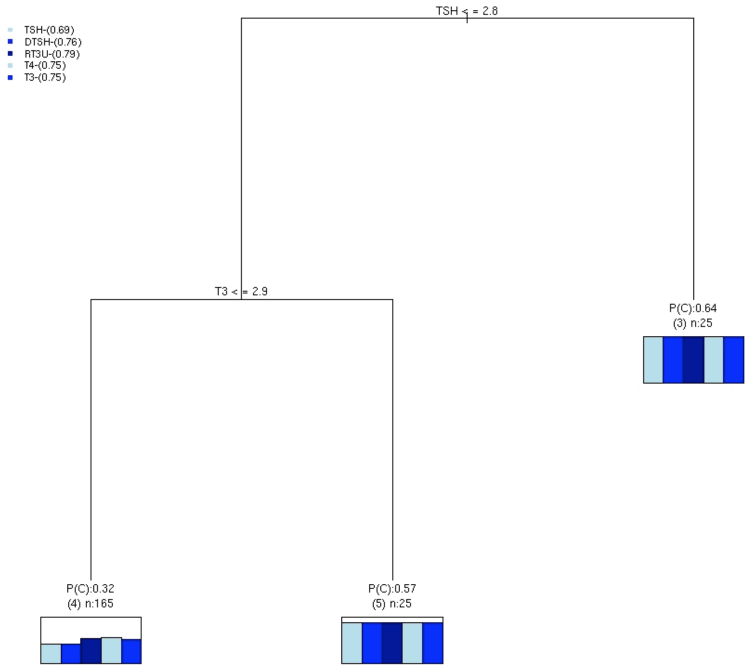


Table 21: Thyroid analysis global MCT misclassification table.

	Terminal Node		
	3	**4**	**5**
Hypothyroid	24	6	0
Hyperthyroid	0	11	24
Normal	1	148	1

7.1.5 Local MCT

Local MCTs find a more complicated tree than global MCTs, identifying 3 splits or 4 groups (Figure 61). These groups are clearly observed within the ACM images and MDS plots, (Figure 62) and this improved resolution is also obvious in the terminal probabilities of the corresponding tree (Figure 63), which are significantly greater than for global MCTs. The local MCT equalled the performance of boosting MCTs, misclassifying 15 observations (Table 22), however the proximity matrices have a more defined structure. However using the ACM, K-means (finding 4 groups) and PAM (finding 3 groups) only misclassified 12 and 10 observations respectively.

Figure 61: Thyroid analysis local MCT RE and AIC plots.

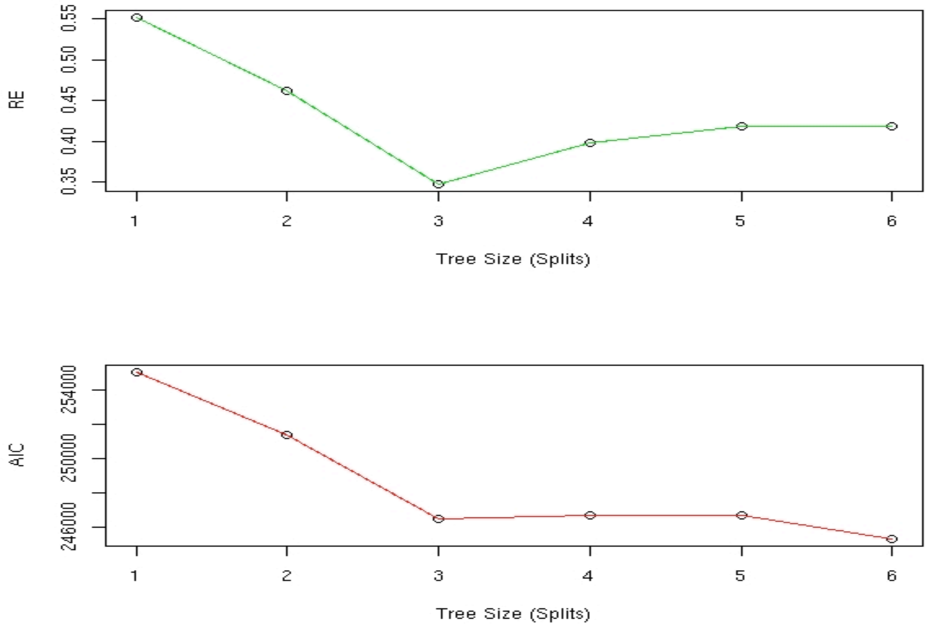


Figure 62: Thyroid analysis local MCT ACM images and MDS plots.

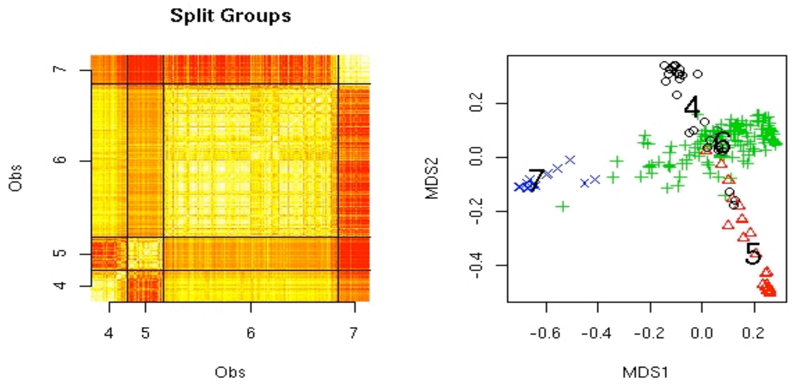


Figure 63: Thyroid analysis local MCT with SSR splitting and GPA consensus combining.

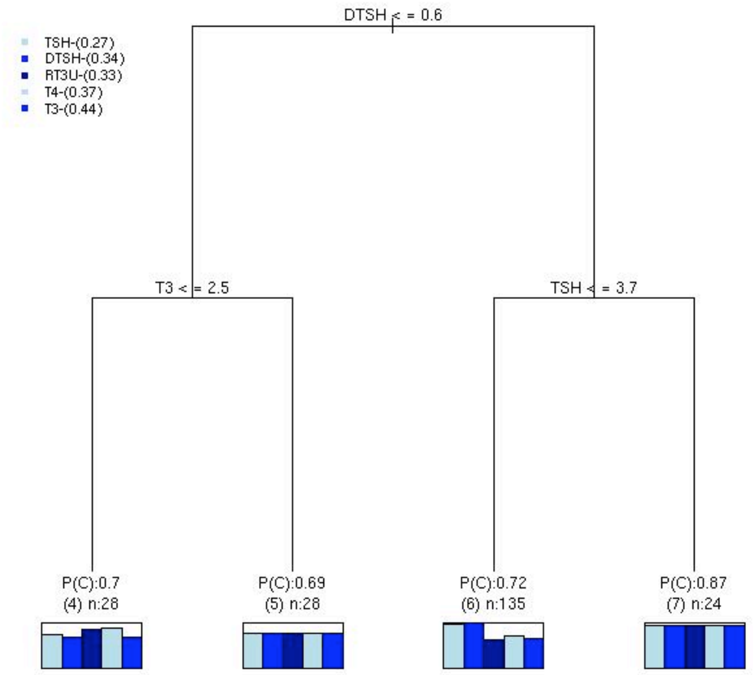


Table 22: Thyroid analysis local MCT misclassification table.

	Terminal Node			
	4	**5**	**6**	**7**
Hypothyroid	0	0	6	24
Hyperthyroid	8	27	0	0
Normal	20	1	129	0

7.1.6 Thyroid summary

A clear result of this example is the marked improvement of general clustering performance that is achieved by using the proximity matrix from either random forests or treeboost. Of the two ensembles the random forest proximity is clearly more stable, and is suited for input into other clustering techniques. The boosted proximity matrix, although producing a more optimal MCT, has a less well defined structure that is not found by other methods.

The fact that K-means and PAM on the proximities do better than trees is primarily due to the fact that trees are constrained by the valid splits available in the variables within the predictor set. When the 15 observations misclassified by MCTs are compared to a classification tree predicting the three groups, which misclassifies 14 observations, it is clear that MCTs are approaching the optimal tree. More so it is clear that the improvements gained by PAM and K-Means are because they are not constrained by the predictor variables.

The differences between the local and global MCTs are expected. Local MCTs have the luxury of removing entire groups, allowing them to focus on groups that may be hard to separate, where as global MCTs are always observing the entire dataset. In this example, the local MCT was more complicated, however more accurate. This accuracy is found not only in the misclassification performances but also in the probability of expression for each terminal node of the tree. Global MCTs found one terminal node that is below random chance expression (3 terminal nodes, random chance expression is P(C)=0.33). By making the tree more complicated the terminal nodes found by local MCTs were all above random chance expression. As a result the predictive performance of local MCTs is improved.

Overall MCT approaches are shown to improve on AA-MRT, AA-RF and equal the performance of AA-Treeboost. However as they are limited by their tree structure, in this analysis MCTs do not perform as well as PAM or K-means. In fact these methods by searching for groups within the consensus matrix, without knowledge of the known labels, perform better than a classification tree. This highlights the quality of the grouping structure within the consensus and at the same time the limits of a simple tree structured for clustering or classification.

7.2 Clustering Categorical Variables: Breast Cancer Dataset

The breast cancer dataset (Wolberg and Mangasrian 1990) contains 699 observations on 11 variables, one being an index variable, 9 being ordered or nominal, and 1 target class (Table 23). This dataset was sourced from the "*mlbench*" R package (Leisch and Dimitriadou 2005). The aim of this study is to present and compare performances of all tree-based methods for clustering categorical data. For a fair performance comparison the data will be divided in two with 349 training set observations and 350 test set observations.

Firstly the base tree methods are presented. These are Db-MRT on the Gower distance, and MRTs, random forests and treeboost on the binary substituted form of the response. As the response dataset is the binary substituted dataset, these models are not auto-associative. Therefore through this section the random forest and treeboost methods are referred to as binary substituted random forest and binary substituted treeboost. Secondly, the results for local and global MCTs are presented. Finally a summary of the methods and comparison of the results is presented.

Table 23: Breast cancer analysis dataset description.

Variable Name	**Description**	**Type**
Id	*Sample code number*	Character
Cl.thickness	*Clump Thickness*	Ordinal {1 to 10}
Cell.size	*Uniformity of Cell Size*	Ordinal {1 to 10}
Cell.shape	*Uniformity of Cell Shape*	Nominal {1 to 10}
Marg.adhesion	*Marginal Adhesion*	Nominal {1 to 10}
Epith.c.size	*Single Epithelial Cell Size*	Ordinal {1 to 10}
Bare.nuclei	*Bare Nuclei*	Ordinal {1 to 10} -16 Missing
Bl.cromatin	*Bland Chromatin*	Nominal {1 to 10}
Normal.nucleoli	*Normal Nucleoli*	Nominal {1 to 10}
Mitoses	*Mitoses*	Nominal {1 to 10}
Class	*Cancer classification*	Nominal {benign, malignant}

7.2.1 Gower dissimilarity Db-MRT

From the RE curve of the Db-MRT (Figure 64) it clear that only two groups have been identified. From the MDS scatter plot (Figure 65) of the distance matrix only two groups found by the tree can be observed. Observation of the misclassification table for the tree in Table 24 show these to groups correspond well with the benign and malignant breast cancers with a misclassification rate of 8 % on the external test set.

Figure 64: Breast cancer analysis Gower distance Db-MRT RE curve.

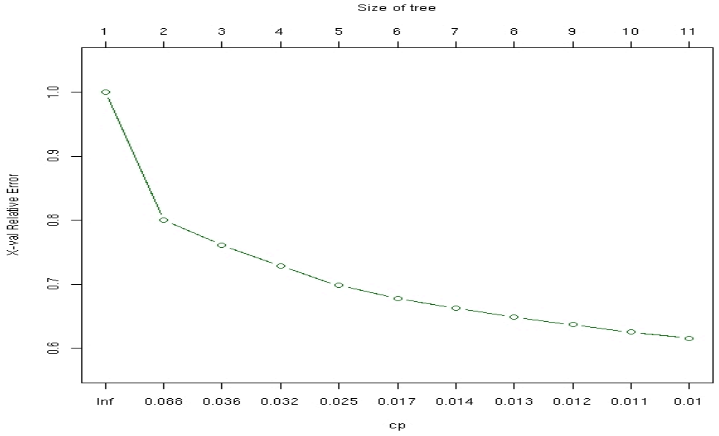
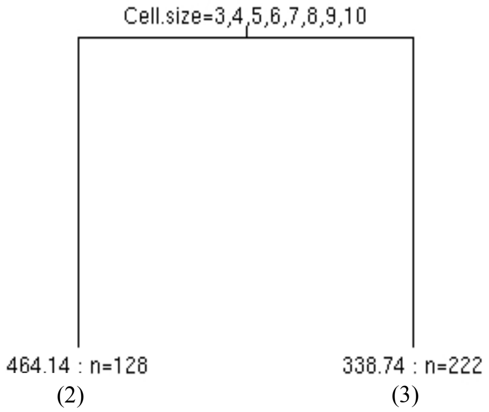
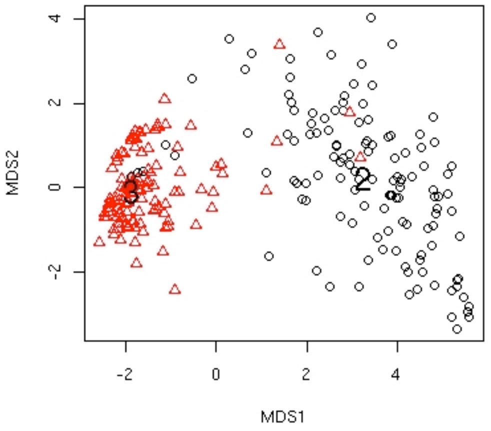


Figure 65: Breast cancer analysis Gower distance Db-MRT.

<table>
<tr><th>Gower Distance Db-MRT</th><th>MDS terminal node location plot using the Gower Distance Matrix</th></tr>
<tr><td>


</td><td>


</td></tr>
</table>

7.2.2 Binary substituted MRT

From the RE curve of the binary substituted MRT (Figure 66) it clear that only two groups have been identified. From the MDS scatter plot (using a Euclidean distance between observations within the response matrix) (Figure 67) only the two groups found by the tree can be observed. Observation of the misclassification table for the tree in Table 24 show these to groups correspond well with the benign and malignant breast cancers with a misclassification rate of 8 % on the external test set.

Figure 66: Breast cancer analysis binary substituted MRT RE curve.

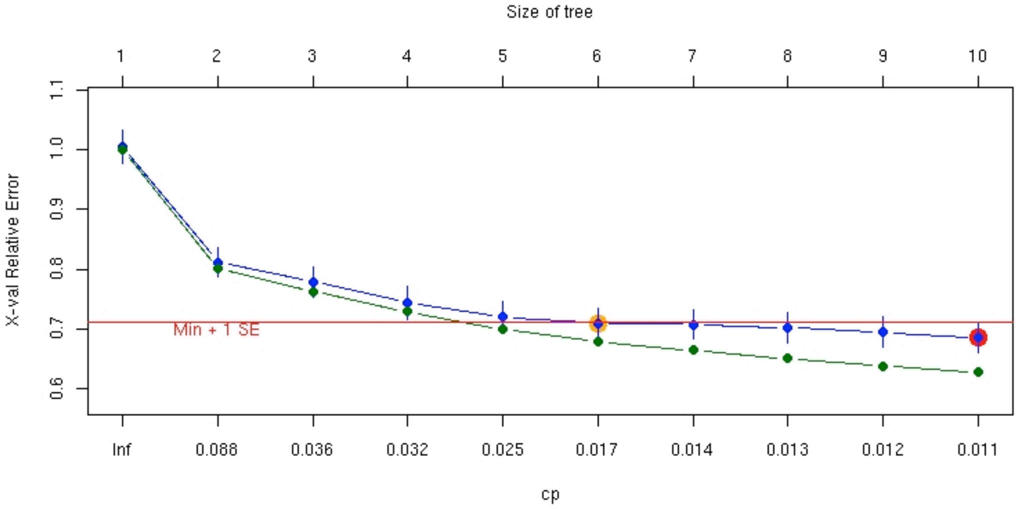
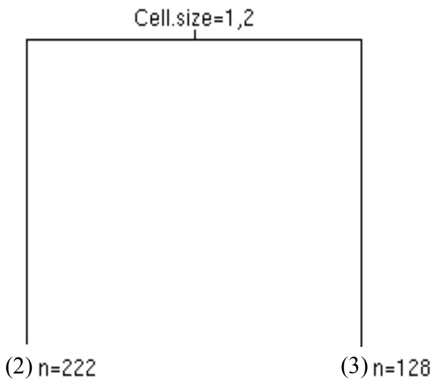
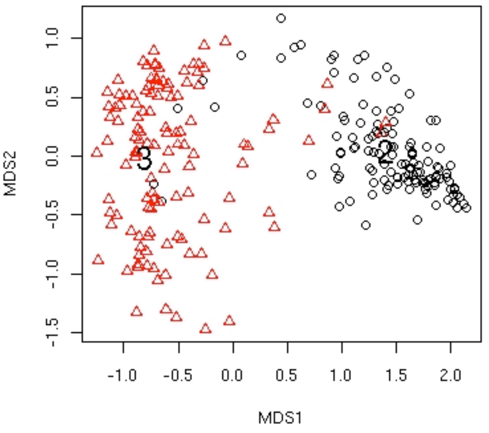


Figure 67: Breast cancer analysis binary substituted MRT and MDS plot.

Binary Substitution MRT	**MDS terminal locations plot using a Euclidean distance over the binary substitution**
	

7.2.3 Binary substituted RF

Binary substituted RF parameters are set to be the following:

(a) A separate random forest test set of 70 training set observations is removed before the analysis to tune the model.

(b) The bootstrapped sample that is used to grow each tree consists of 196 observations and 3 variables.

(c) A maximum tree size of 10 splits within the forest is allowed.

(d) The minimum terminal node size for each tree within the forest is 10 observations.

(e) There are 200 trees within the random forest.

Figure 68: Breast cancer analysis binary substituted AA-RF error convergence plot.

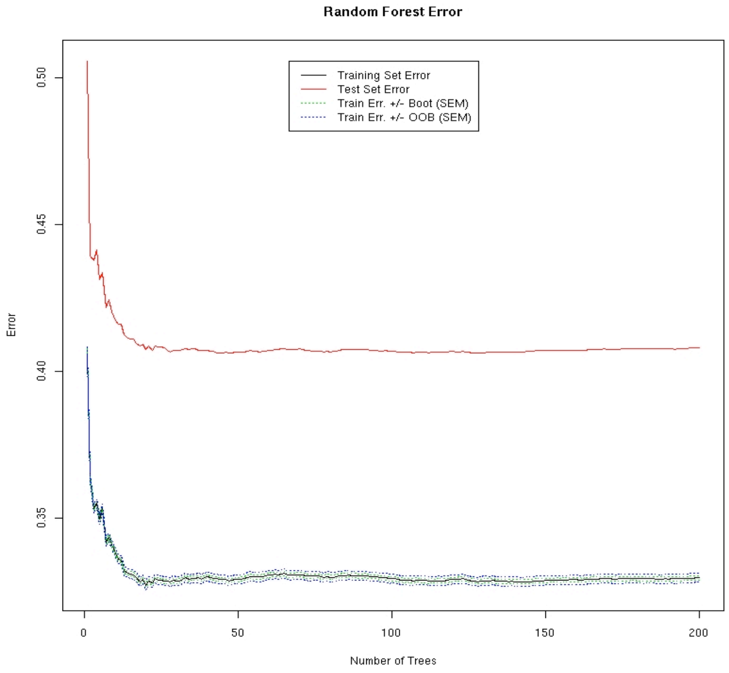


Figure 69: Breast cancer analysis binary substituted RF RE curves.

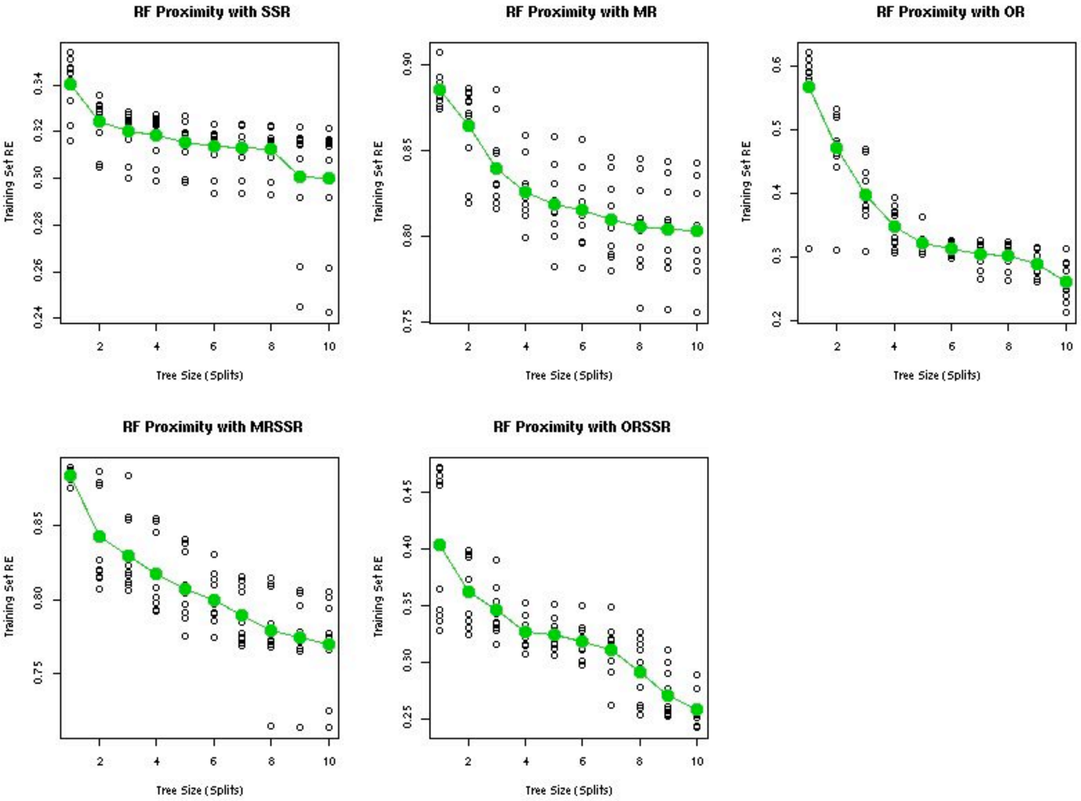


Figure 70: Breast cancer analysis binary substituted random forests MCT built with SSR splitting to 2 splits.

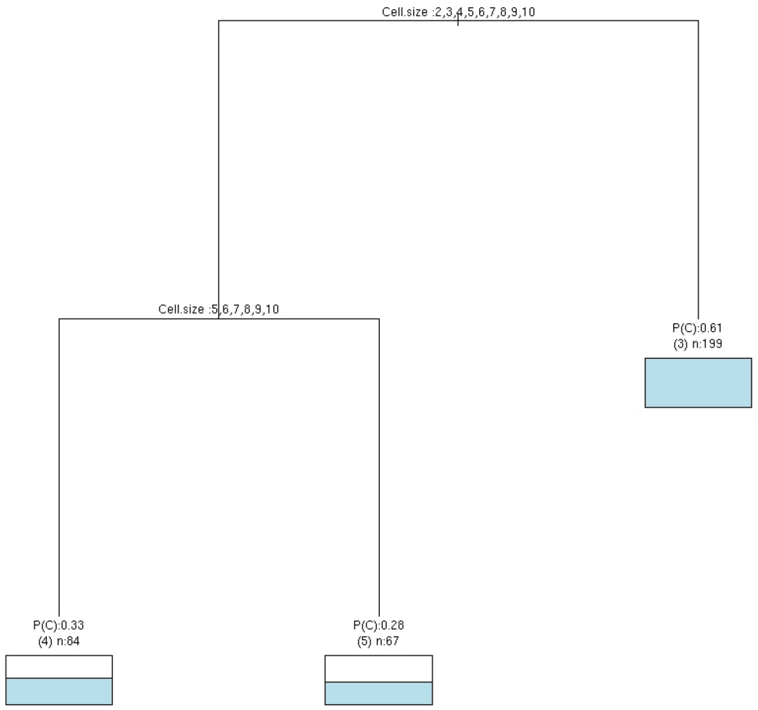
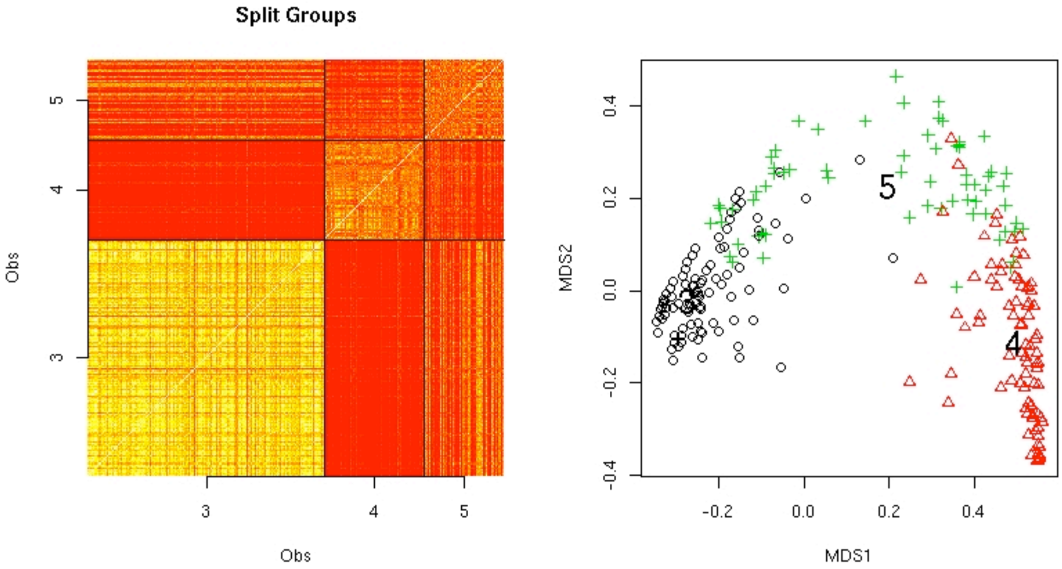


Figure 71: Breast cancer analysis binary substituted RF proximity images.



From the error convergence plot (Figure 68) it is obvious that the random forests error is stable at 200 trees. The RE curves of the MCT splitting criteria however are less clear (Figure 69). Here SSR splitting is selected at two splits, as the RE is stable at approximately 0.32 between 2 and 8 splits. This is not the case with the other splitting criteria. The tree (Figure 70) and the corresponding random forest proximity images (Figure 71) indicate a high certainty in terminal node 3, however markedly less certainty is terminal nodes 4 and 5. This is reflected in the misclassification table (Table 24) with terminal node 3 clearly being the most accurate.

7.2.4 Binary substituted treeboost

The boosted set of trees is grown using the following parameters:

(a) A separate random forest test set of 70 training set observations is removed before the analysis to tune the model.

(b) The bootstrapped sample that is used to grow each tree consists of 196 observations and 3 variables.

(c) A maximum tree size of 2 splits within the boosting is allowed.

(d) The minimum terminal node size for each tree within the boosting is 10 observations.

(e) There are 500 trees within the boosted set.

(f) Shrinkage Parameter set at 0.05.

Figure 72: Breast cancer analysis binary substituted treeboost error convergence plot.

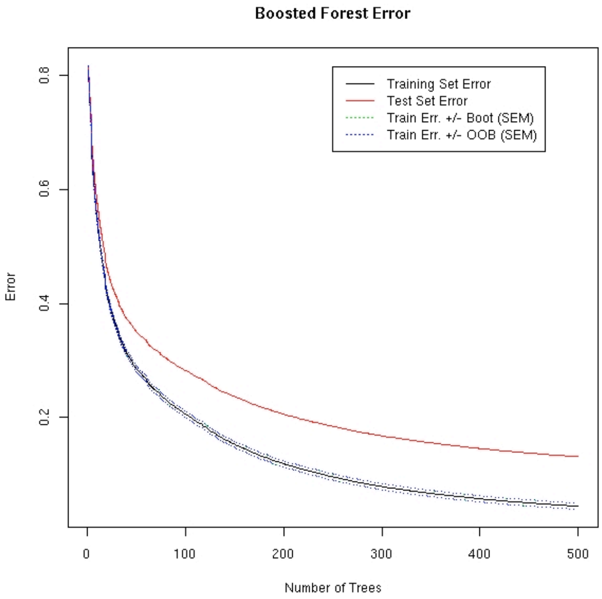


Figure 73: Breast cancer analysis binary substituted treeboost RE curves.

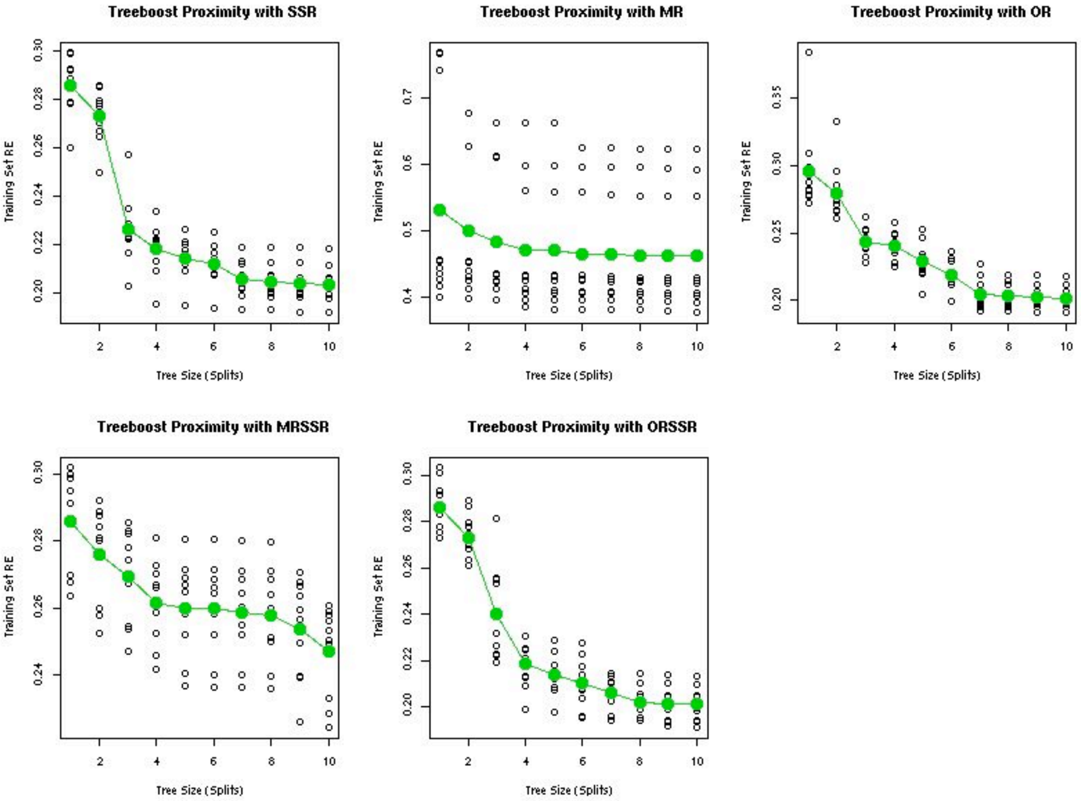


Figure 74: Breast cancer analysis binary substituted treeboost MCT built with SSR splitting to 3 splits.

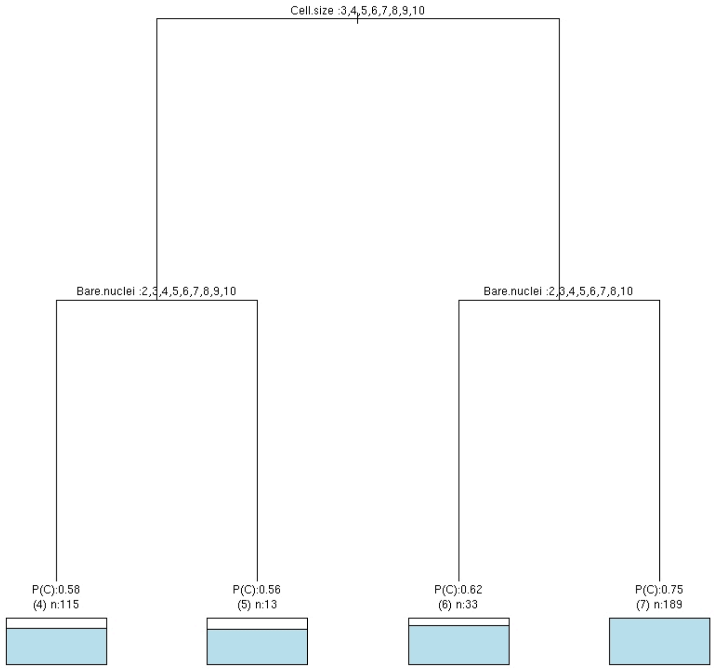
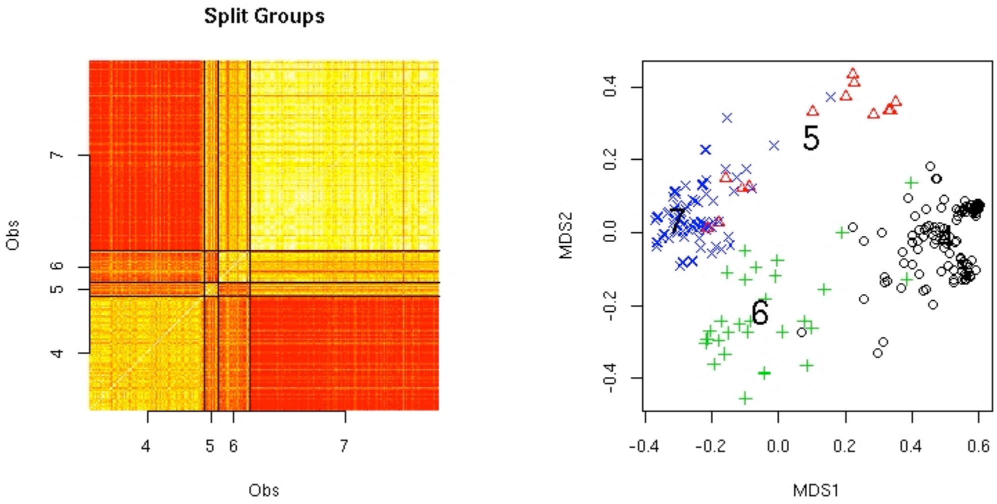

Cell.size :3,4,5,6,7,8,9,10
Bare.nuclei :2,3,4,5,6,7,8,9,10
Bare.nuclei :2,3,4,5,6,7,8,10
P(C):0.58
(4) n:115
P(C):0.56
(5) n:13
P(C):0.62
(6) n:33
P(C):0.75
(7) n:189

Figure 75: Breast cancer analysis binary substituted treeboost proximity images.



From the error convergence plot (Figure 72) it can be seen that the treeboost model has stabilised after 500 trees. The RE curves (Figure 73) show that the splitting functions SSR, OR and OR-SSR each pick a tree size of 3 splits. Of these SSR is selected, as the cross-validated performances are the most stable at a RE of approximately 0.22. From the tree (Figure 74) and proximity images (Figure 75) a high level of certainty exists throughout each terminal node. This is mirrored in the misclassification table (Table 24), where an error rate of 6.85 % is observed on the test set.

7.2.5 Base method misclassification results

Table 24: Breast cancer analysis misclassification performances of base methods.

Method	Tree Node	Training Set		Test Set	
		Benign	Malignant	Benign	Malignant
Gower DB-MRT (7.5 % Error)	**2**	215	7	202	5
	3	18	110	23	119
	% Error	*7.73 %*	*5.98 %*	*10.36 %*	*3.9 %*
BS-MRT (7.5 % Error)	**2**	7	215	5	202
	3	110	18	119	23
	% Error	*5.98 %*	*7.73 %*	*3.9 %*	*10.36 %*
RF (10.44% Error)	**3**	197	2	183	2
	4	3	81	2	92
	5	33	34	40	30
	% Error	*15.45 %*	*0.89 %*	*0.89 %*	*25.8 %*
Treeboost (6.5 % Error)	**4**	10	105	10	110
	5	8	5	13	9
	6	27	6	24	5
	7	188	1	178	0
	% Error	*4.2 %*	*10.26 %*	*4.44 %*	*9.4 %*

The single tree results highlight the similarities between binary substitution and the Gower dissimilarity, as BS-MRT and Gower Db-MRT produced the same terminal nodes but with an exactly opposite tree and show marked similarities in the MDS locations plots (Figure 65, Figure 67). The consensus based methods show the same first split using variable “Cell.size” as the single tree methods, however binary substituted RF shows a different decision point to the treeboost (Figure 70, Figure 74).

Interestingly both consensus based methods find more complex trees, however only in the case of binary substituted treeboost does this translate into improved performance. Surprisingly binary substituted RF performs worst of all other methods (Table 24). By observation of the misclassification tables it is clear that binary substituted RF is strongly biased towards the malignant group in the training set, to the detriment of overall predictive performance. The best performing model of the base methods is clearly treeboost, with the lowest classification error and a clear proximity image.

7.2.6 Global MCT

Global MCT random forests are grown on each variable in the training set with the following parameters:

(a) A separate random forest test set of 70 training set observations is removed before the analysis to tune the model.

(b) The bootstrapped sample that is used to grow each tree consists of 196 observations and 3 variables.

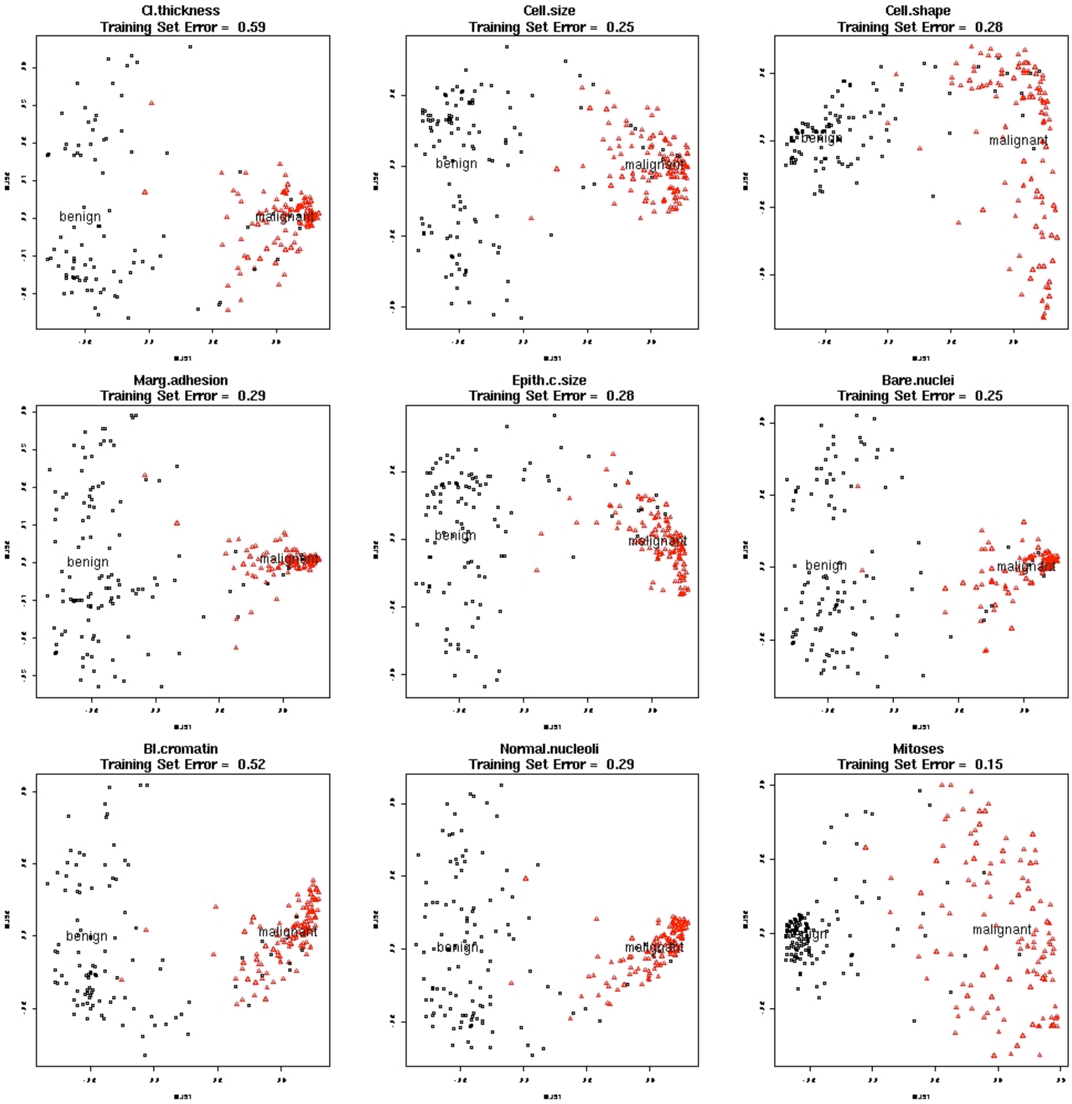
(c) A maximum tree size of 10 splits within the forest is allowed.

(d) The minimum terminal node size for each tree within the forest is 10 observations.

(e) There are 200 trees within the random forests.

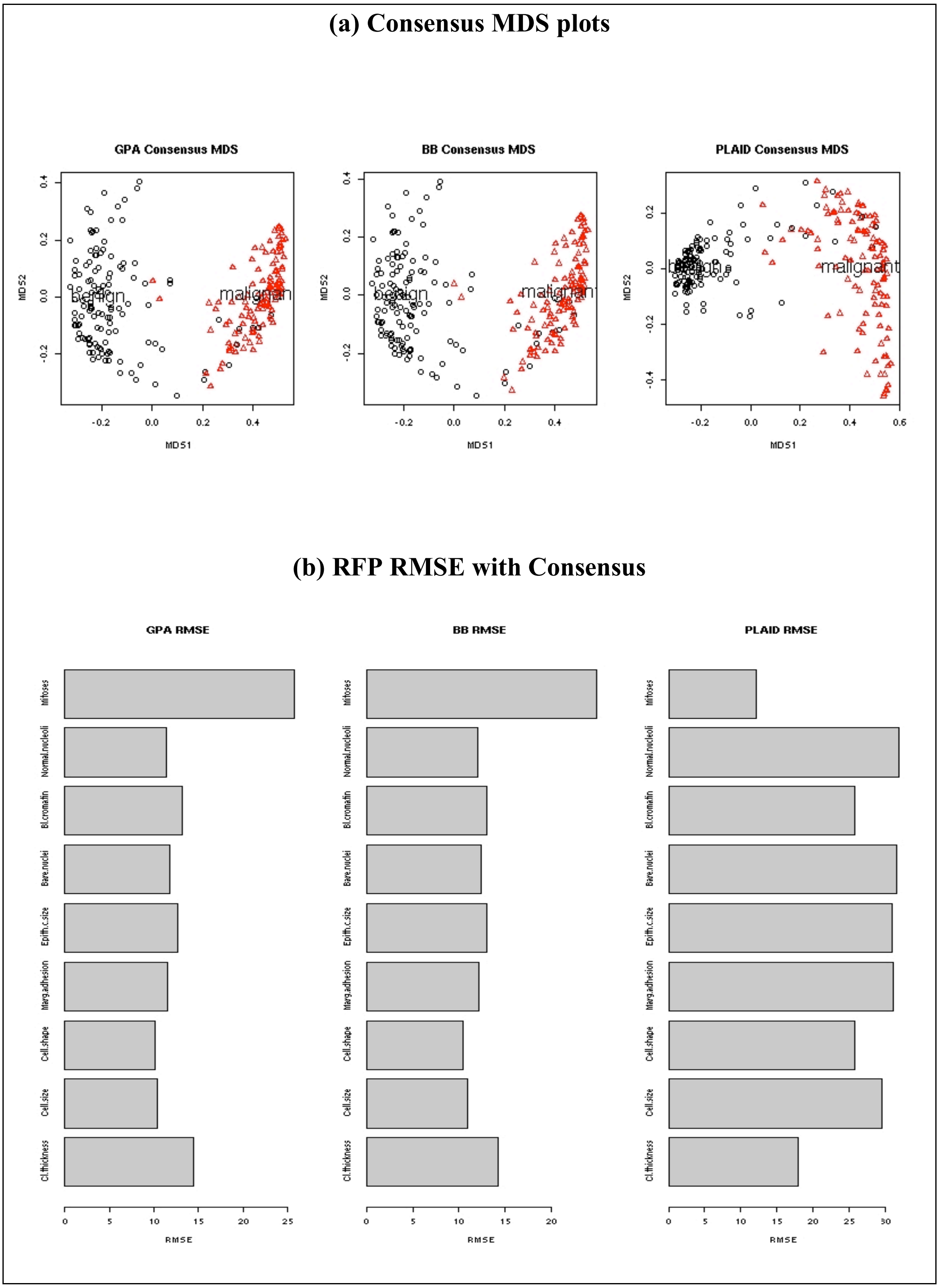
The individual RFPs (Figure 76) clearly show that the forests are finding a clear distinction between benign and malignant cancer groups. The training set performance for predicting each variable by the forest as a misclassification error is printed in the plot titles.

Figure 76: Breast cancer analysis individual RFP MDS plots.



Each RFP combination method is used to construct a consensus matrix (Figure 77(a)). By observation of the MDS plots, it appears that BB and GPA find similar configurations, and PLAID finds a different structure. This observation is reinforced by RMSE plots between the individual RFPs and the consensus (Figure 77(b)). From the RMSEs it can be seen that BB and GPA clearly favour the middle variables, performing poorest on Mitoses and Cl.thinckness, whereas PLAID favours these variables at the expense of the others.

Figure 77: Breast cancer analysis consensus MDS plots and RMSEs.



(a) Consensus MDS plots

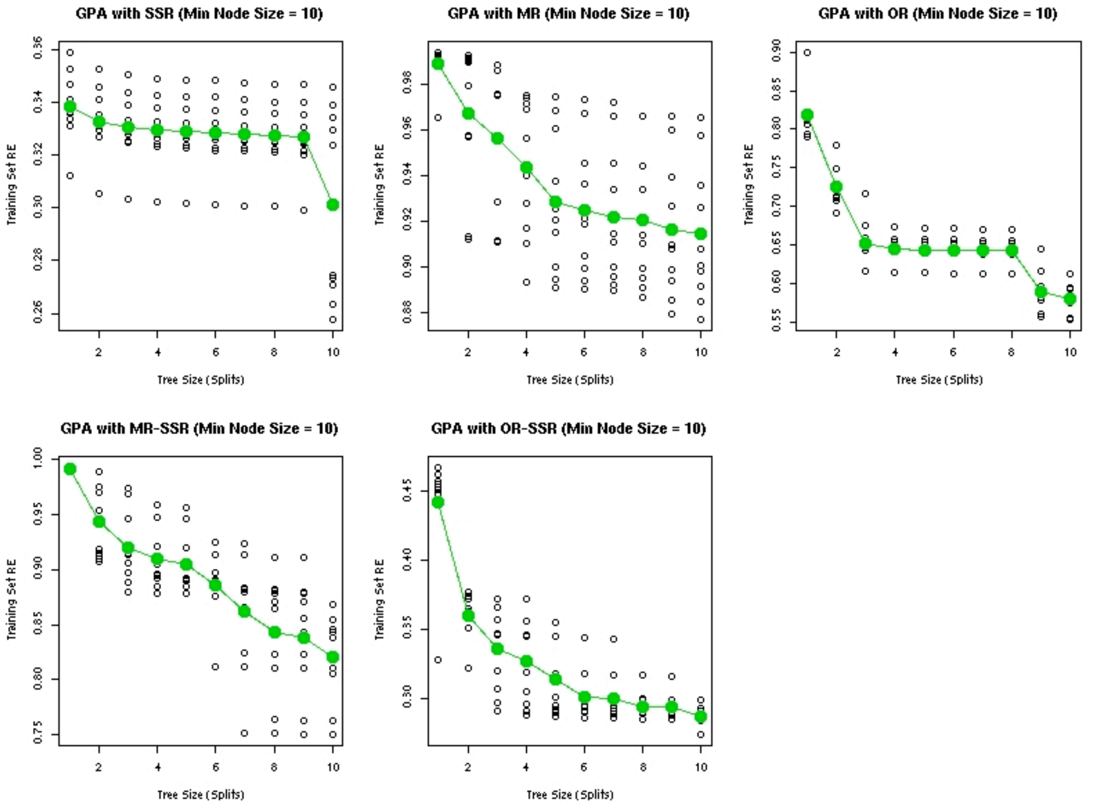
(b) RFP RMSE with Consensus

To assess how far to grow the MCT 10-fold CV is performed on the consensus matrix, with a minimum terminal node size fixed to 10 observations (Figure 78).

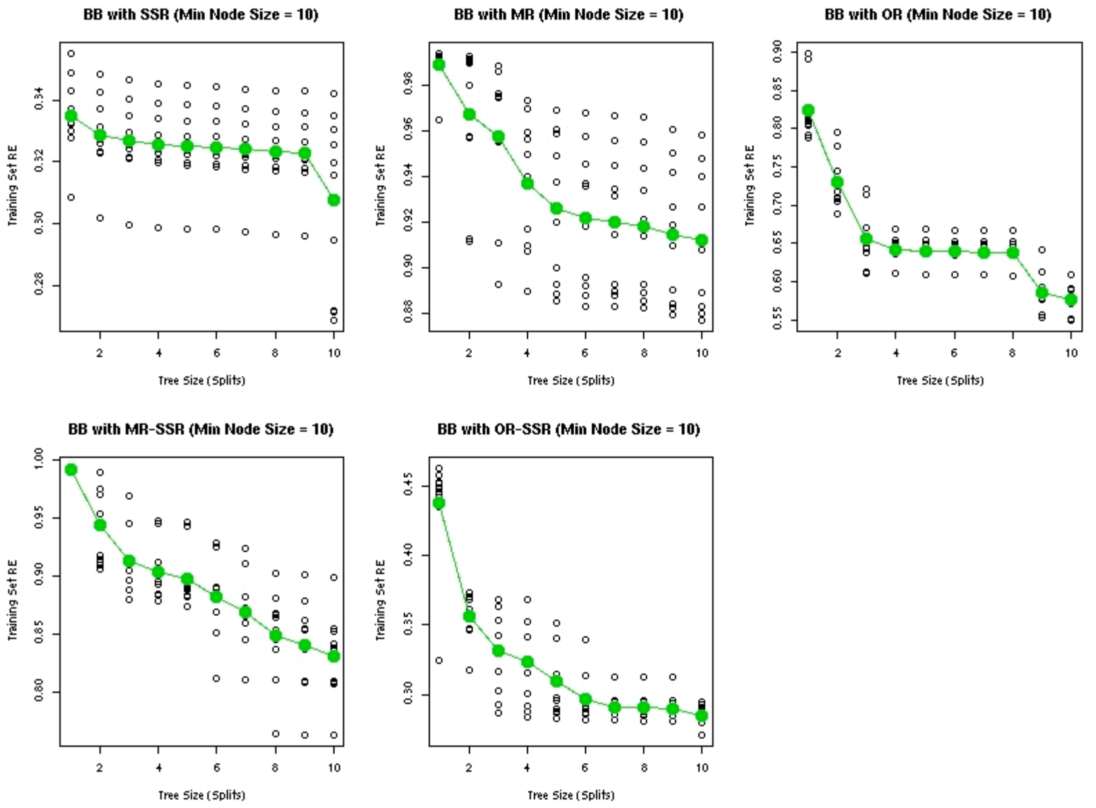
Clearly the best structure is resolved by the splitting method OR-SSR, which finds a RE elbow for GPA (Figure 78a) and BB (Figure 78b) at 6 splits, and for PLAID (Figure 78c) at 5 splits, with a corresponding mean RE of between 0.2 and 0.25. This RE equates to an R^2 of approximately 0.7, meaning the predicted consensus matrix accounts for between 65 % and 75 % of the consensus variation.

Figure 78: Breast cancer analysis global MCT 10-fold CV RE curves.

(a) GPA

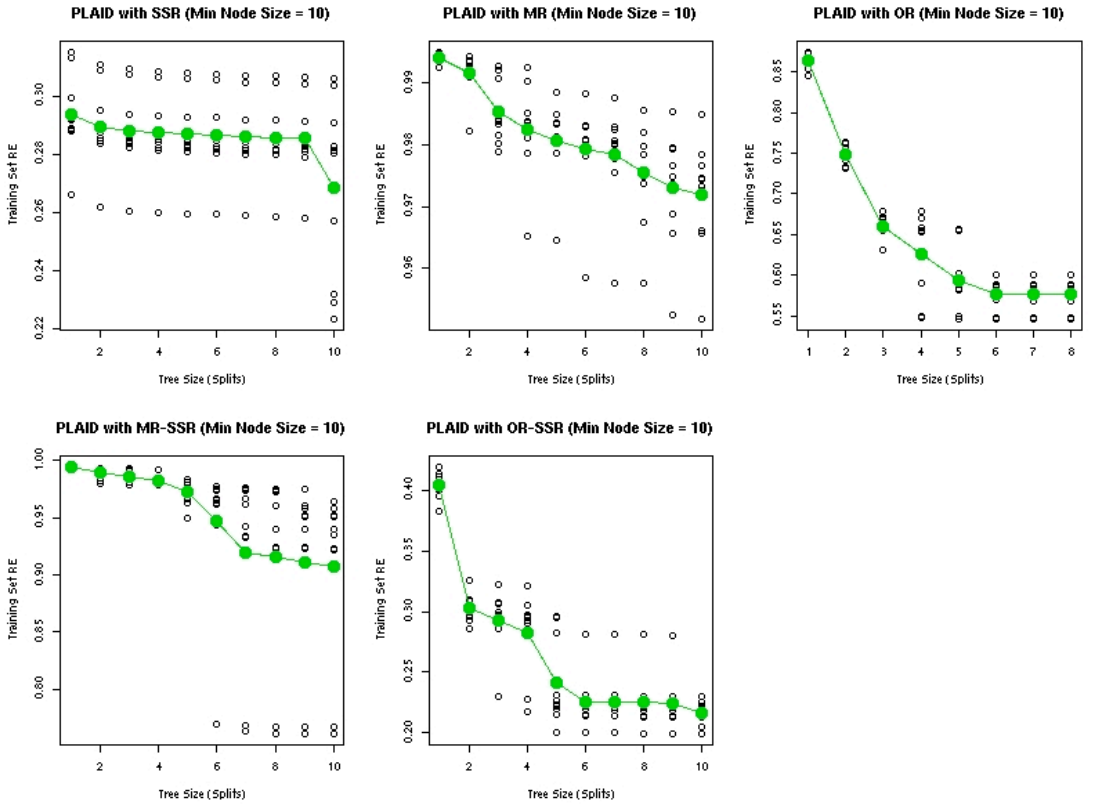


(b) BB



BB with SSR (Min Node Size = 10)
Training Set RE
0.28
0.30
0.32
0.34
2
4
6
8
10
Tree Size (Splits)
BB with MR (Min Node Size = 10)
Training Set RE
0.88
0.90
0.92
0.94
0.96
0.98
2
4
6
8
10
Tree Size (Splits)
BB with OR (Min Node Size = 10)
Training Set RE
0.55
0.60
0.65
0.70
0.75
0.80
0.85
0.90
2
4
6
8
10
Tree Size (Splits)
BB with MR-SSR (Min Node Size = 10)
Training Set RE
0.80
0.85
0.90
0.95
1.00
2
4
6
8
10
Tree Size (Splits)
BB with OR-SSR (Min Node Size = 10)
Training Set RE
0.30
0.35
0.40
0.45
2
4
6
8
10
Tree Size (Splits)

(c) PLAID



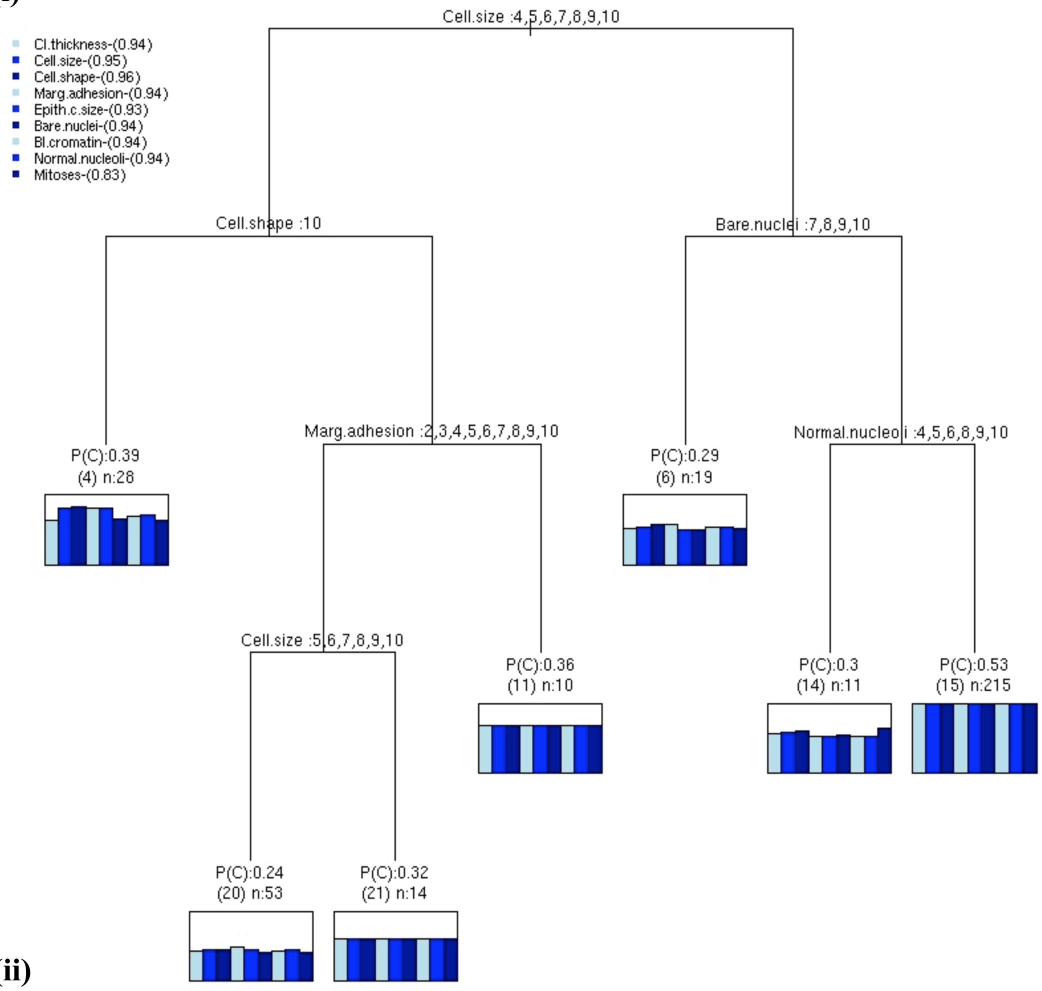
PLAID with SSR (Min Node Size = 10)
Training Set RE
0.22
0.24
0.26
0.28
0.30
2
4
6
8
10
Tree Size (Splits)
PLAID with MR (Min Node Size = 10)
Training Set RE
0.96
0.97
0.98
0.99
2
4
6
8
10
Tree Size (Splits)
PLAID with OR (Min Node Size = 10)
Training Set RE
0.55
0.60
0.65
0.70
0.75
0.80
0.85
1
2
3
4
5
6
7
8
Tree Size (Splits)
PLAID with MR-SSR (Min Node Size = 10)
Training Set RE
0.80
0.85
0.90
0.95
1.00
2
4
6
8
10
Tree Size (Splits)
PLAID with OR-SSR (Min Node Size = 10)
Training Set RE
0.20
0.25
0.30
0.35
0.40
2
4
6
8
10
Tree Size (Splits)

The global MCTs for GPA and BB are grown to 6 splits (7 clusters) and PLAID to 5 splits (6 clusters) all using the OR-SSR splitting method. The trees for GPA and BB are the same (Figure 79(a,i)) and only show subtle differences from splits observed in the PLAID MCT (Figure 79(b,i)). The terminal node locations found by the trees are displayed on the MDS plots of the consensus matrices (Figure 79(a,ii) and Figure 79(b,ii)) for GPA, BB or PLAID respectively. From this it can be seen that most effort is spent identifying the malignant group, with the majority of the benign group being positioned in both trees in terminal node 15. In the left corner of the MCTs, the response variable importance list (YVIP) list can be found. The structure found in the YVIP matches the RMSE combination plots (Figure 77b). The mean of the consensus at each terminal node is printed below the terminal node as a probability, "P(C)", along with the terminal node number in brackets, and the number of training set observations within that node. A bar plot of the P(C)s of each individual RFP at each terminal node is also presented.

Figure 79: Breast cancer analysis best global MCTs and terminal node location MDS plots.

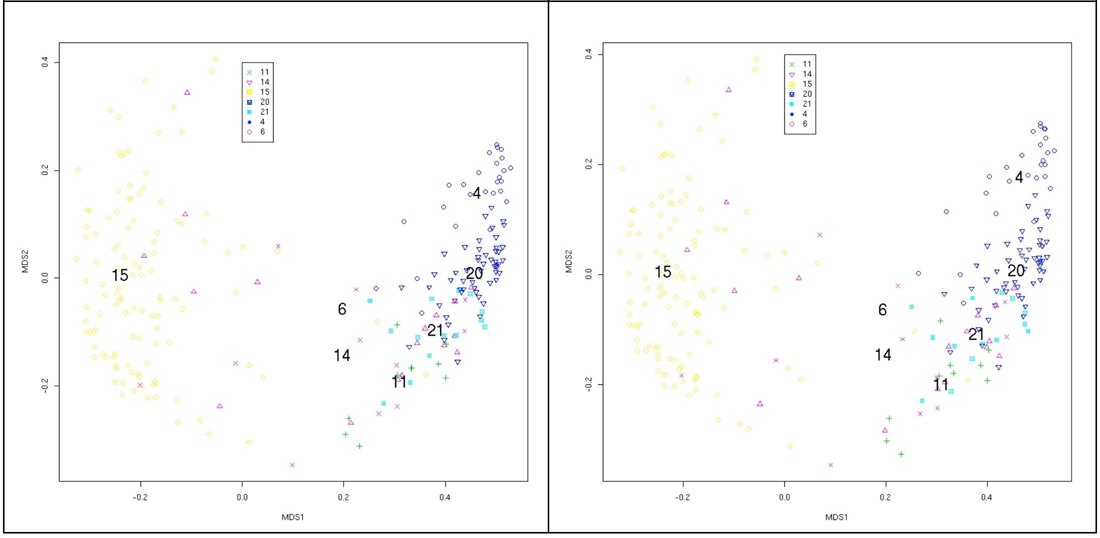
(a) GPA & BB OR-SSR MCT Tree

(i)



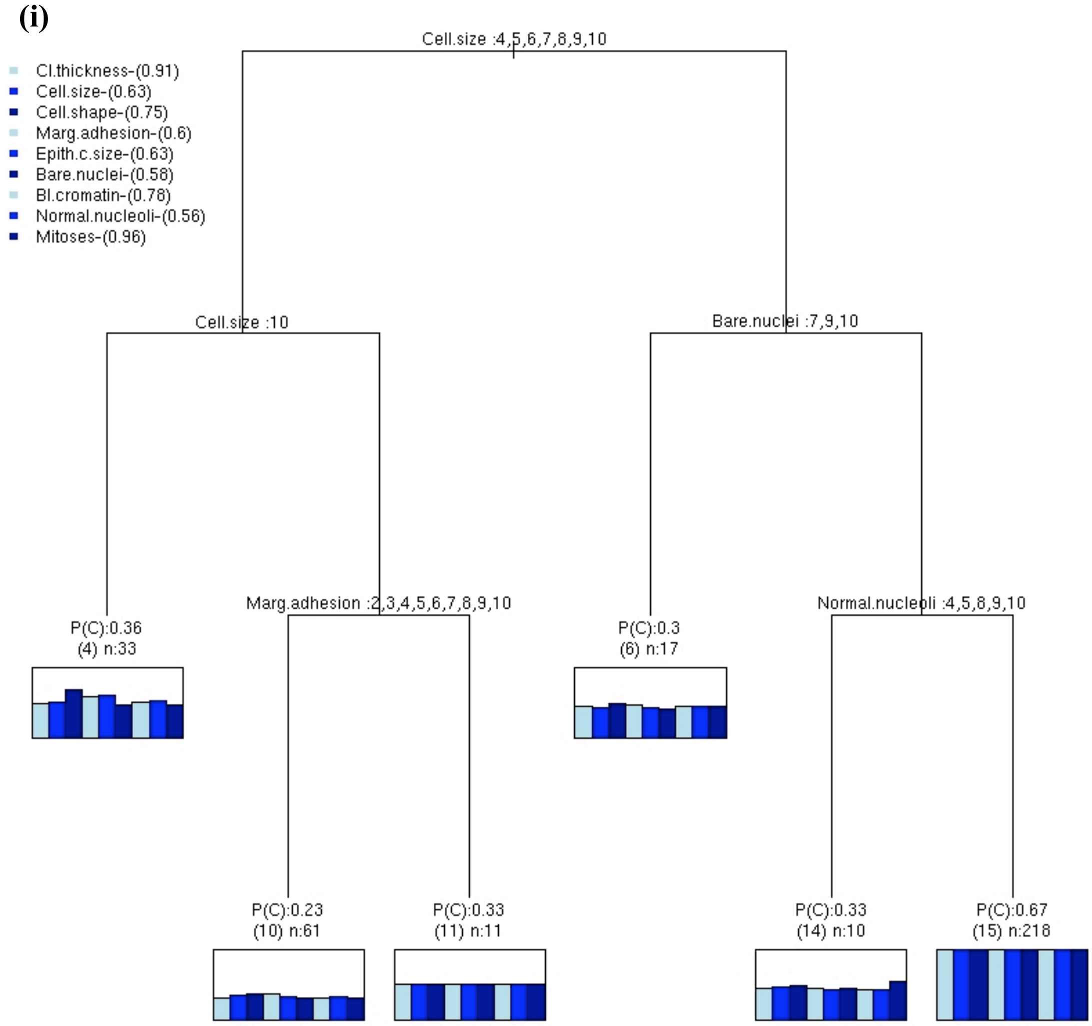
(ii)

GPA MCT Terminal Node Locations	BB MCT Terminal Node Locations

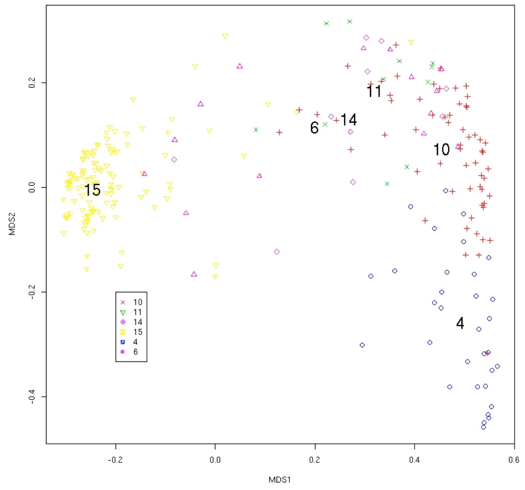


(b) PLAID OR-SSR MCT Tree

(i)



(ii) **Plaid MCT Terminal Node Locations**



The training and test set performances for global MCTs (Table 25) show an overall test sample misclassification rate of approximately 4.9 %. When compared to supervised classification on the same data, a decision tree grown to 4 splits gives test set misclassification rate of 6.3 % and a random forest gives a test set misclassification rate of 2.8 %. Therefore the performance of global MCTs for the breast cancer dataset is approaching that of a random forest.

Table 25: Breast cancer analysis global MCT misclassification performances.

(a) GPA & BB (6 % Overall Error)

MCT Node	**Train Set (6.3 % misclassification)**		**Test Set (4.9 % misclassification)**	
	Benign	**Malignant**	**Benign**	**Malignant**
4	0	28	0	30
6	6	13	7	8
11	3	7	2	5
14	5	6	0	3
15	213	2	213	5
20	2	51	1	60
21	4	10	2	13
Overall Misclassification	*8.6 %*	*1.7 %*	*5.33 %*	*4 %*

(b) PLAID (5.4 % Overall Error)

MCT Node	**Train Set (6 % misclassification)**		**Test Set (4.87 % misclassification)**	
	Benign	**Malignant**	**Benign**	**Malignant**
4	0	33	0	34
6	6	11	7	8
10	6	55	3	70
11	3	8	2	4
14	3	7	0	3
15	215	3	213	5
Overall Misclassification	*8.0 %*	*2.6 %*	*5.33 %*	*4 %*

7.2.7 Local MCT

As local MCTs build a separate forest at each node, the random forest parameters are presented in percentages. The local MCT parameters are set at the following:

(a) The bootstrapped sample used to grow each tree within each node is defined as 70% of node observations and 33% of variables.

(b) Maximum random forest tree size is 3 splits.

(c) Minimum MCT and random forest terminal node size is 10 observations.

(d) Random forest size is 200 trees.

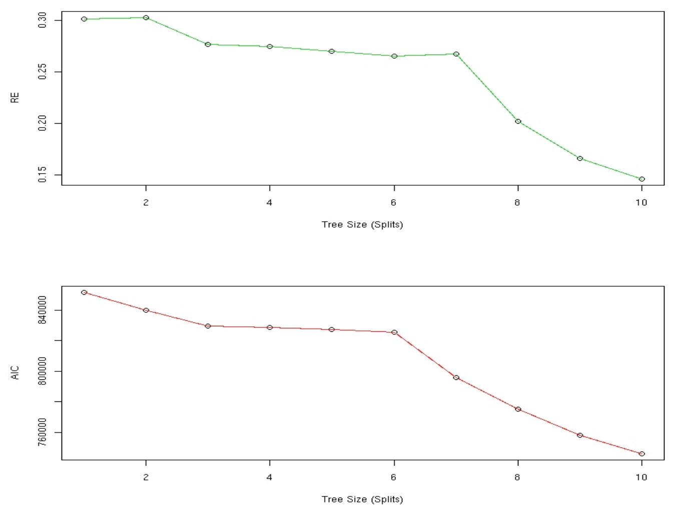
(e) OR-SSR splitting criteria.

RE and AIC are generated to assess local MCT tree size. For a fair comparison with global MCTs only OR-SSR splitting criteria is employed as it clearly outperformed other splitting criteria in global MCTs for this problem. Local MCTs are run using all three RFP combination methods.

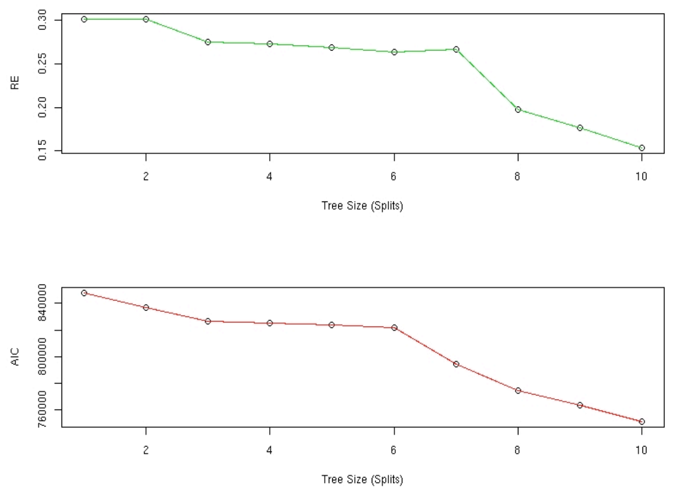
For local MCTs the results for each combination method with OR-SSR splitting are identical. The RE and AIC plots each indicate a tree size of 3 split or 4 groups (Figure 80) and the resulting MCT tree for each combination method at 3 splits is the same (Figure 81i). This results in the same misclassification performance of 5.44 % error on the test set (Table 26). The only difference in the trees is the subtle differences observed in the PLAID MDS plot of the ACM matrix when compared to either the BB or GPA plots (Figure 81ii).

Figure 80: Breast cancer analysis local MCT RE and AIC plots.

GPA



BB



PLAID

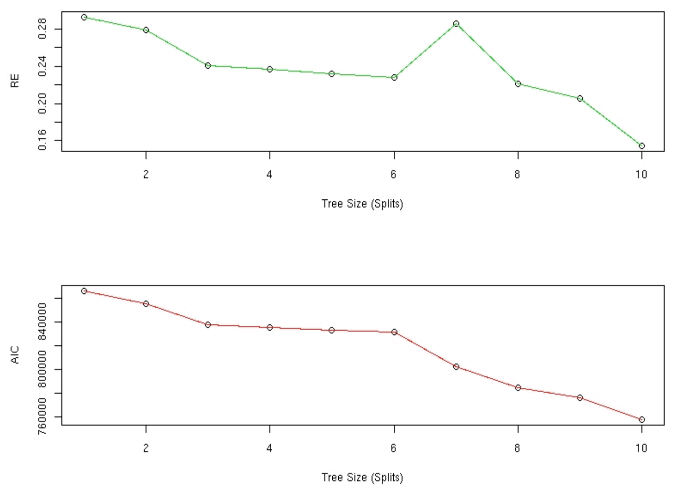
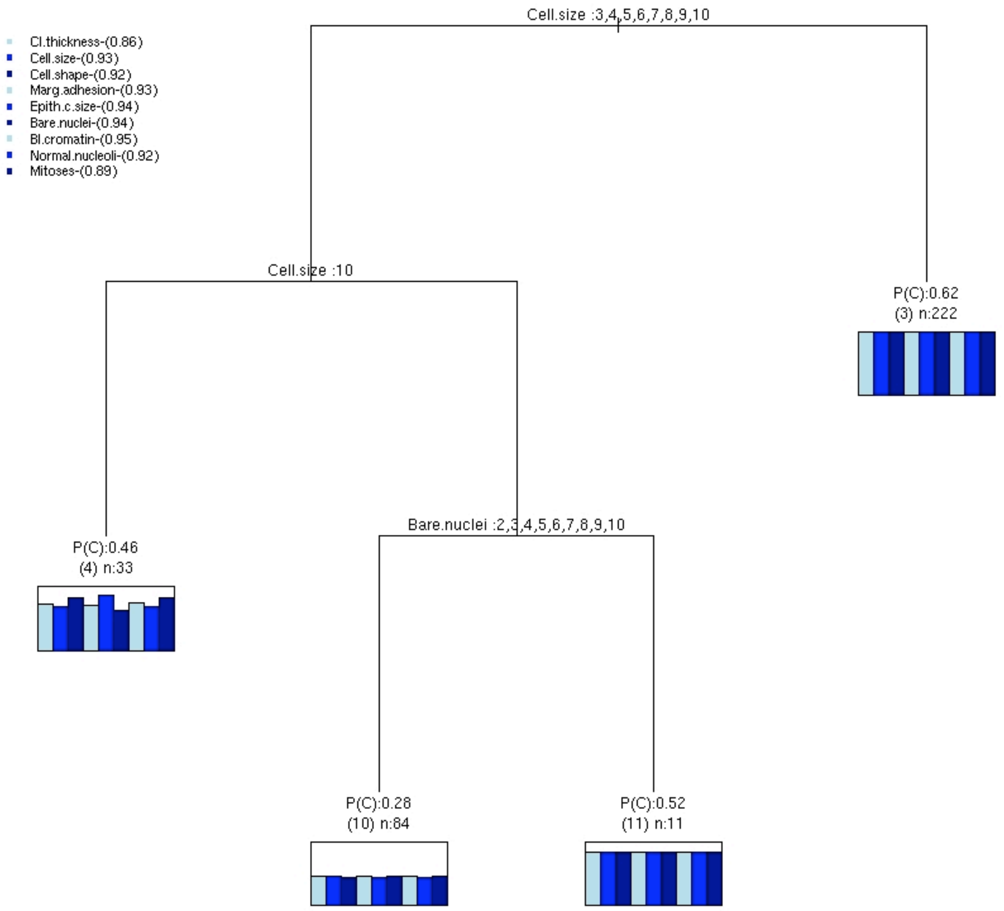


Figure 81: Breast cancer analysis GPA, BB and PLAID, OR-SSR local MCT.

(i) GPA, BB and PLAID local MCT



(ii) MDS Terminal Node Location Plot

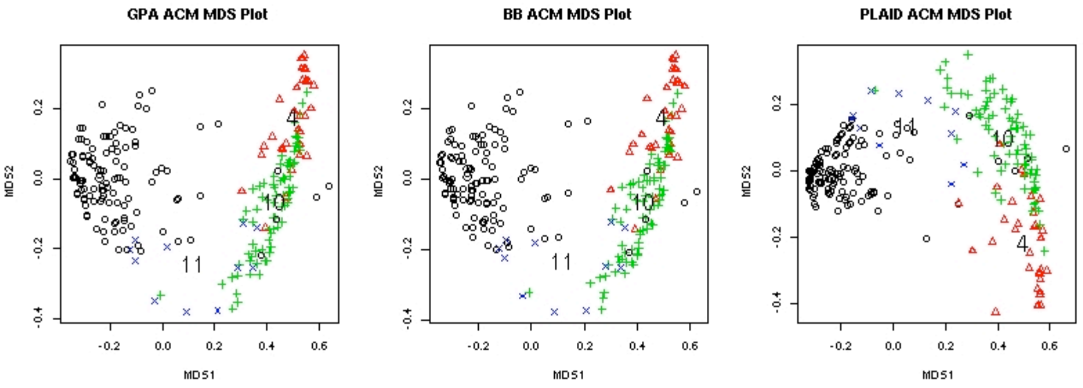


Table 26: Breast cancer analysis local MCT misclassification performances.

MCT Node (5.57 % Total Error)	**Train Set (5.71 % misclassification)**		**Test Set (5.44 % misclassification)**	
	Benign	**Malignant**	**Benign**	**Malignant**
3	215	7	202	5
4	0	33	0	34
10	10	74	10	81
11	8	3	13	4
Overall Misclassification	*4.29 %*	*8.54 %*	*4.44 %*	*7.26 %*

7.2.8 Breast cancer summary

Of all the methods presented, global MCTs using the PLAID consensus produced the most accurate tree (test set misclassification performance 4.87 %) (Table 25). Furthermore, the performances of all MCT methods are better than any base tree method. Compared to existing literature on this dataset MCTs are performing comparably. Clustering using SOM achieved a misclassification rate of 4.68 % (Pantazi, Kagolovsky and Moehr 2002) however this method assumes all variables are ordinal and provides no measures of variable importance. Supervised analysis of this dataset has been shown to perform well below 10 % misclassification, with a linear programming approach achieving 3 % misclassification (Mangasrian and Wolberg 1990).

Given that there are two groups (benign and malignant) within the dataset, the most accurate models in this case were Gower Db-MRT and binary substituted MRT as they found 2 terminal nodes. As accuracy increased so did tree size with the most accurate MCT identifying 6 groups within the data. This inflation of group number is due to the overlap between the two groups. This is reinforced by observation of the consensus MDS images, as in all plots expect AA-Treeboost two groups are obvious.

These results imply that the simple base methods (Gower Db-MRT and binary substituted MRT) do not have sufficient power to identify the overlapping groups.

The improvement in resolution gained from a local MCT should also be noted. All combination methods for local MCTs produced the same tree. Furthermore a smaller tree is obtained with comparable predictive performance. These results highlight the differences between local and global MCTs, and show that once tuned local MCTs produce a more accurate result.

7.3 Mixed Type Profiling: Horse Colic Dataset

In this analysis MCTs are used as a mixed type profiling tool. Here there is no known set of groups to compare against, and therefore the quality of the groups found must be assessed on how representative they are of each response variable. This analysis is performed on the horse colic dataset, where the response set comprises of variables that describe the observed physical state of each horse, and the predictor set are variables that describe the type, site and severity of their colic lesion (Mcleish and Cecile 1989). The goal is to use MCTs to identify groups in the predictor variables describing the lesions that correspond to groups within the response set of physical descriptors.

The horse colic dataset contains 300 observations on 17 variables, 5 being quantitative and 12 being either ordinal or nominal (Table 27). With such a complicated response set spanning many types, it is expected that some variables will display different group profiles. In this analysis MCTs are used as a search for subgroups of response variables that display a common group structure. To do this a recursive search for common group structure using plaid combining is described. The result of this search is subgroups of response variables that have similar configurations within their RFPs. Upon these subgroups, separate MCTs are grown and compared to the structure found in an overall MCT involving all RFPs. This is a data reduction step that is aimed at improving the understanding of group structure within each variable and how it relates to the overall structure within the dataset.

Table 27: Horse colic analysis dataset description.

Variable Set	Variable Name	Description	Type	Missing Values
Response	REC.TEMP	*Rectal temperature*	Continuous	60
Response	PULSE	*Pulse rate*	Continuous	24
Response	CELL.VOL	*Packed cell volume*	Continuous	29
Response	TOT.PROT	*Total protein*	Continuous	33
Response	RESP.RATE	*Respiratory rate*	Continuous	58
Response	TEMP.EXT	*Temperature of extremities*	Ordinal {4 levels}	56
Response	PERIF.PU	*Peripheral pulse*	Ordinal {4 levels}	69
Response	MUCOUS.M	*Mucous membranes*	Nominal {6 levels}	47
Response	CAPILL.R	*Capillary refill time*	Ordinal {2 levels}	34
Response	PAIN	*A subjective judgment of pain level*	Nominal {5 levels}	55
Response	PERISTAL	*Peristalsis*	Nominal {4 levels}	44
Response	ABDOM.DI	*Abdominal distension*	Ordinal {4 levels}	56
Response	NASO.REF	*Nasogastric reflux*	Ordinal {4 levels}	106
Predictor	LESION	*Is surgery required on the lesion*	Dichotomous Yes or No	0
Predictor	LESION.S	*Site of the lesion*	Nominal 1. Gastric 2. Small intestine 3. Large colon 4. Large colon and cecum 5. Cecum 6. Transverse colon. 7. Retum/descending colon 8. Uterus 9. Bladder 10. All intestinal sites 11. None	0
Predictor	LESION.T	*Type of the lesion*	Nominal 1. Simple 2. Strangulation 3. Inflammation 4. Other	60
Predictor	LESION.A	*Subtype of the lesion*	Nominal 1. Mechanical 2. Paralytic 3. N/A	1

7.3.1 MRT methods

To begin analysis on the horse colic dataset, simple MRTs with the Gower distance matrix and binary substituted response sets are grown. If the grouping structure within the response is strong then these methods will adequately describe the groups present. However it is expected that with such a complicated response these methods will be insufficient and unable to find meaningful structure.

7.3.1.1 Gower dissimilarity Db-MRT

Figure 82: Horse colic analysis Gower Db-MRT RE curve.

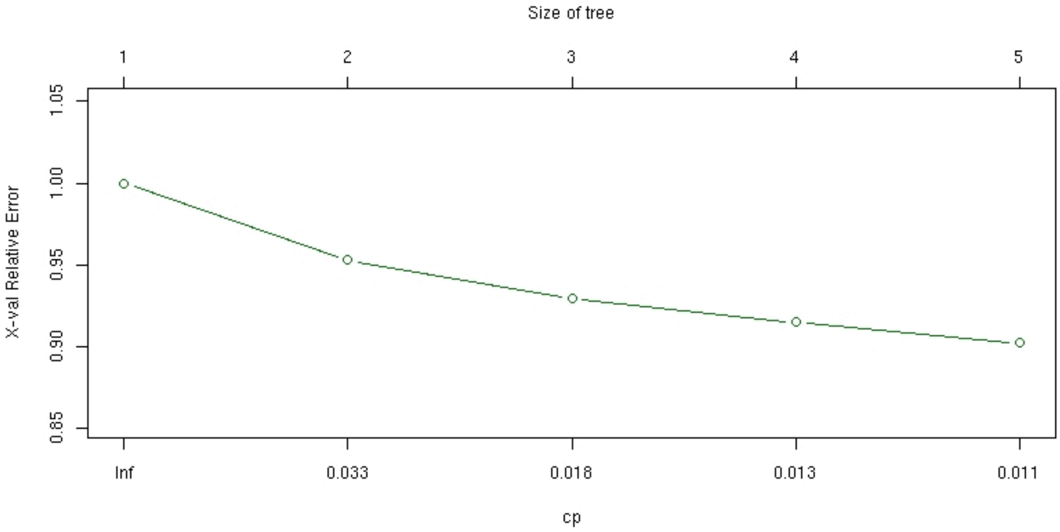
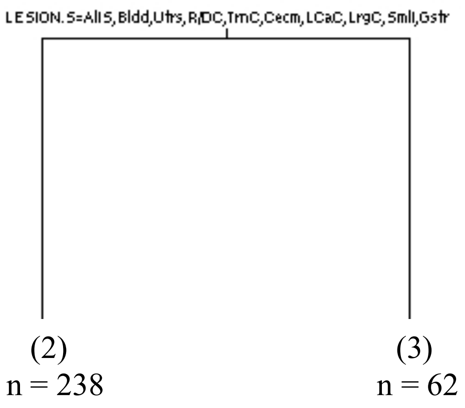
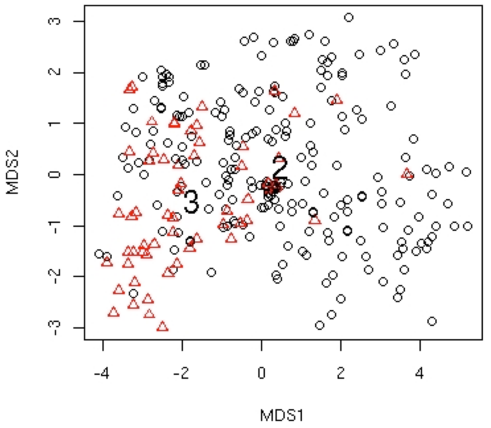


Figure 83: Horse colic analysis Gower Db-MRT and terminal node locations.

Gower Db-MRT	**MDS plot of the Gower distance matrix.**
	

7.3.1.2 Binary substituted MRT

Figure 84: Horse colic analysis binary substituted MRT RE curve.

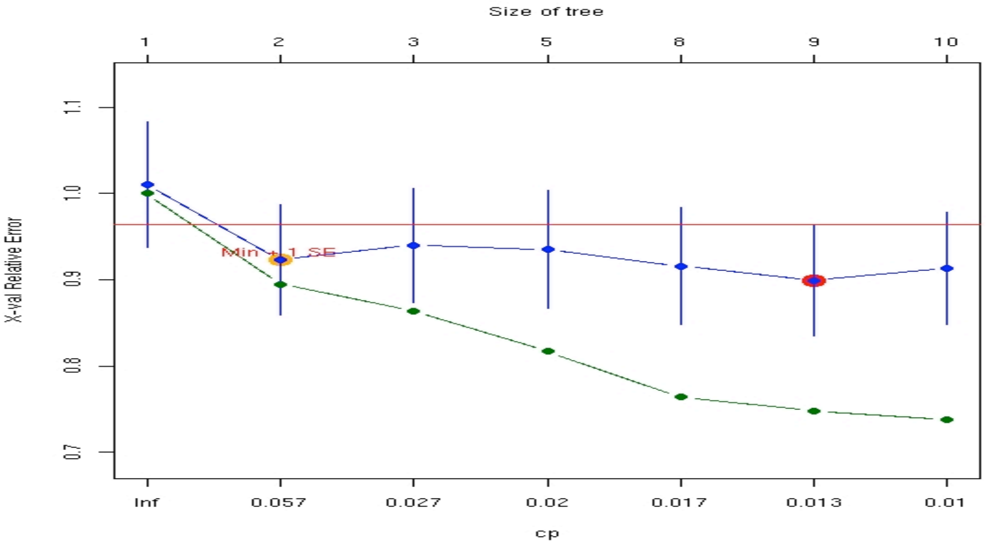
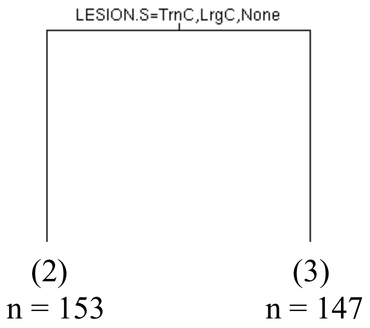
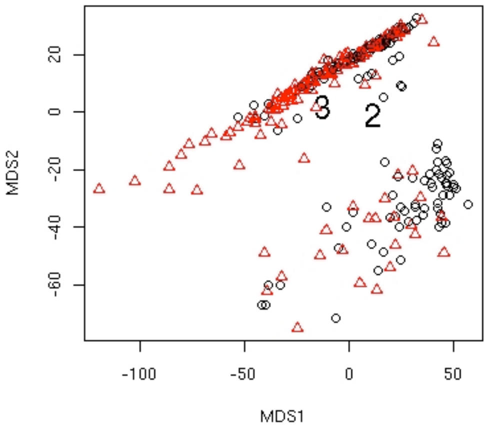


Figure 85: Horse colic analysis binary substituted MRT and terminal node locations.

Binary Substituted MRT	**MDS plot of the binary substituted response using a Euclidean distance.**
	

7.3.1.3 MRT method summary

The issue of missing values within the response set is primary when interpreting the MRT methods. The Gower distance simply ignores comparisons that involve a missing value in its distance computation. The result of such an approach is no observable grouping structure within the response set (Figure 83). This lack of structure is represented by a high RE of 95 % (Figure 82) and results in a simple single split tree (Figure 83).

For the binary substituted data, the missing values are imputed using a K-nearest neighbour averaging on a Euclidean distance (Hastie, Tibshirani, Narasimhan and Chu 2005). The effect of this is a more obvious grouping structure within the MDS plot, which is not found by the MRT (Figure 85). The MRT itself acknowledges this with a poor predictive performance, displaying a RE of 0.93 +/- 0.052 (Figure 84).

The implications of these results are that the groups within the profiling set are not obvious within the predictor set. As a result the trees found are simple and poorly performing.

7.3.2 Tree-based ensemble methods

To benchmark the MCT methods, overall consensus matrices are produced using random forests and treeboost on the binary substituted response. The important results of these techniques will be observable structure within the MDS plots of the ensemble proximity matrices, and measures of predictive accuracy and stability of tree based methods with the error convergence plots.

The consensus approaches show a much improved resolution of the lesion groups within the response (Figure 89, Figure 93). However the complexity of these relationships is highlighted with the random forest models requiring over 200 trees to become stable and treeboost over 100 (Figure 86, Figure 90). The partitions of the proximity images show for both methods a clear 2-3 group structure (Figure 87, Figure 91).

The MCTs for each ensemble proximity matrix are slightly different (Figure 88, Figure 92) with nodes 6 and 7 being found by lesion type in random forest ensemble MCT splitting and by whether the lesion was surgical or not, in treeboost ensemble MCT. For the split, of the 106 strangulation lesion types in LESION.T, 98 of these are flagged as being surgical in LESION implying a strong overlap between the two potential splits. The total difference between the two splits is 28 observations, which is 9.34 % of the observations.

7.3.2.1 Binary substituted random forests

Figure 86: Horse colic analysis binary substituted RF error convergence plot.

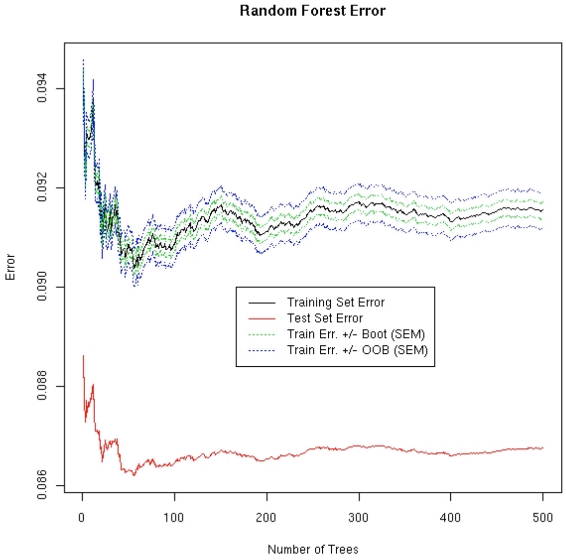


Figure 87: Horse colic analysis binary substituted RF RE curves.

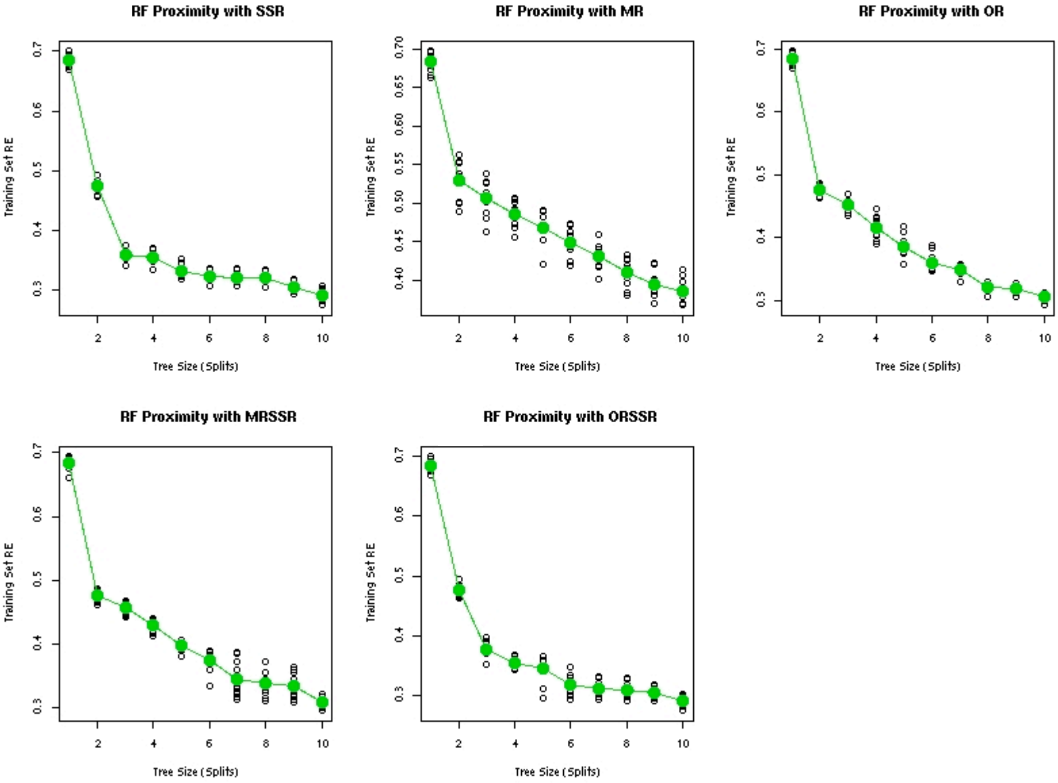


Figure 88: Horse colic analysis RF tree grown to 3 splits using SSR splitting.

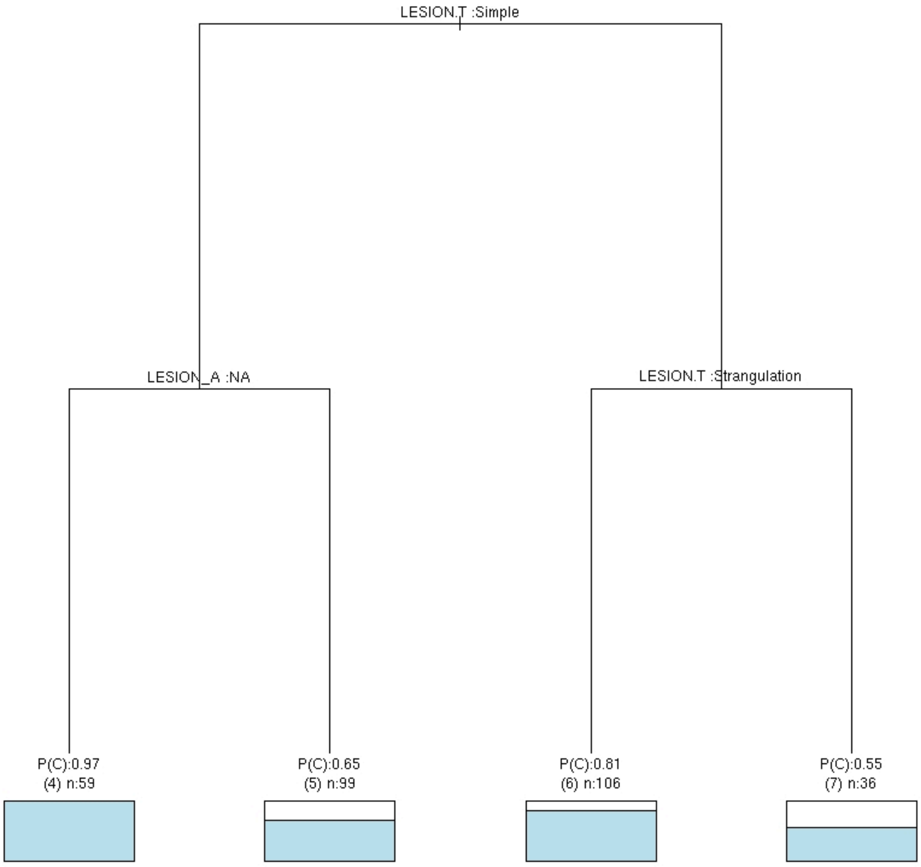
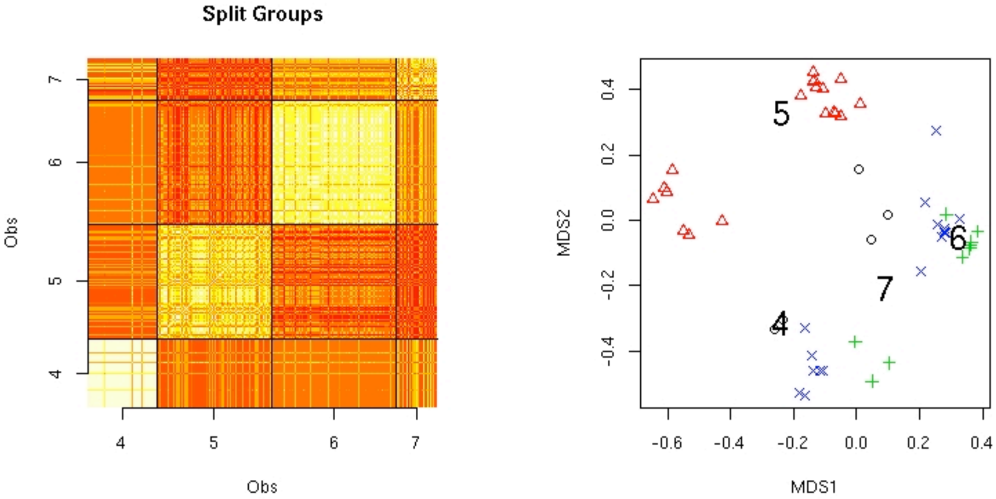


Figure 89: Horse colic analysis RF proximity images.



7.3.2.2 Binary substituted treeboost

Figure 90: Horse colic analysis binary substituted treeboost error convergence plot.

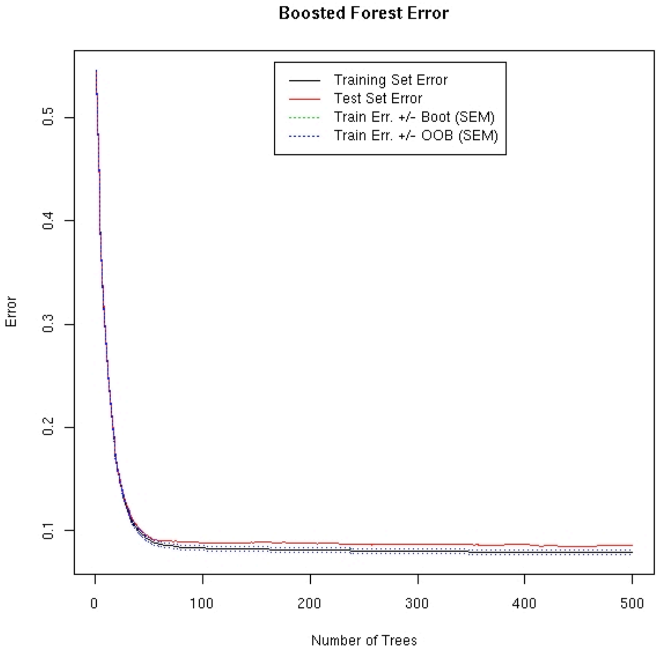


Figure 91: Horse colic analysis binary substituted treeboost RE curves.

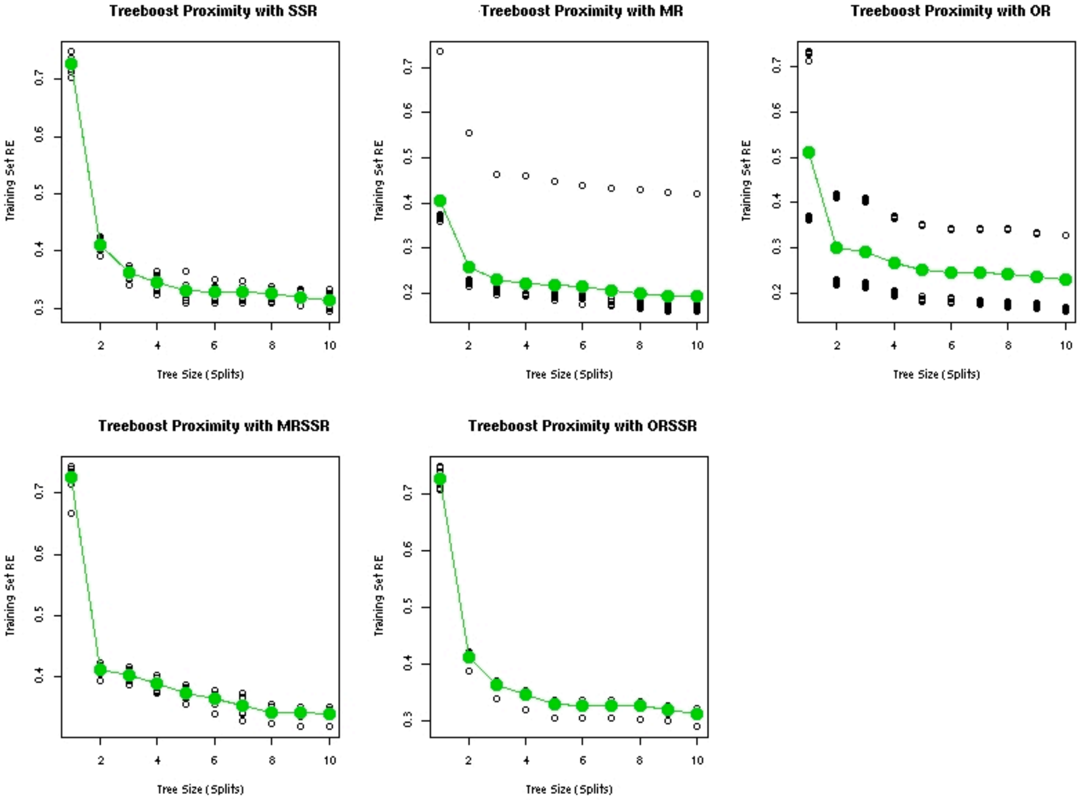


Figure 92: Horse colic analysis treeboost tree grown to 3 splits using SSR splitting.

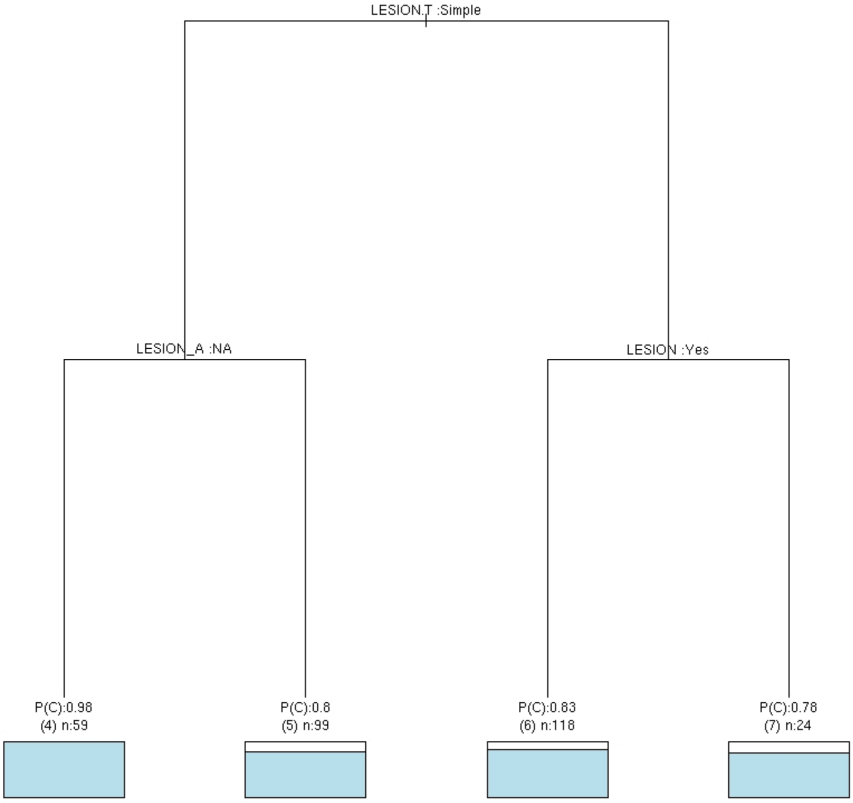
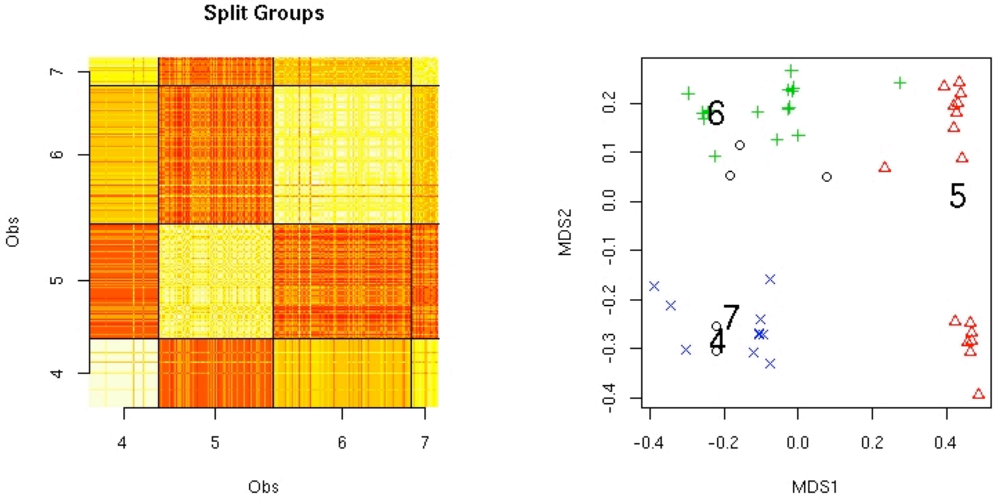


Figure 93: Horse colic analysis treeboost proximity images.



7.3.3 MCT methods

MCTs treat each variable within the response set individually to gain more understanding on the grouping structure of each individual response. As a result MCTs can be used not only for finding common profiles that exist in the entire dataset, but also sub-profiles or groups that are present in only a subset of response variables. This analysis focuses on MCT's ability to find these sub-groups and improved understanding of the final groups gained through the filtering process.

The first step in this analysis is the construction of a global MCT to profile the complete response set of the horse colic dataset. On this terminal node filtering is performed. This will show that not all response set variables express every node within the MCT. Secondly an algorithm of filtering the response variables before an MCT is grown is presented. This algorithm finds groups of variables within the profiling set using the PLAID consensus generation method. By doing this it is shown that further understanding and improved resolution of the groups found by the MCT is possible.

7.3.3.1 *Complete response set global MCT*

The first step in the analysis is to run the random forests for each response separately. These RFPs are used for all analysis. The global MCT random forest parameters:

- Set seed at 123.
- Separate test percentage of 60 observations to evaluate the ensemble's performance.
- 168 observations and 1 predictor used to construct each tree.
- Maximum tree size is 10 splits.
- Minimum terminal node size is 10 observations.
- The random forest is built to 200 trees.

Before being passed into any further analysis the performance of the random forests is assessed. The percent training set error in the title of the RFP images plot in Figure 94 show that for the response variables REC.TEMP, CELL.VOL, TOT.PROT and RESP.RAT the error in prediction is greater than if a simple mean is used as the prediction. As a result these variables are removed from the analysis.

To construct the global MCT the following profiling variables are used:

- PULSE
- TEMP.EXT
- PERIF.PU
- MUCOUS.M
- CAPILL.R
- PAIN
- PERISTAL
- ABDOM.DI
- NASO.REF

The consensus MDS plots (Figure 95a) show that the structure within the individual RFPs (Figure 94) has been maintained. Each combination method appears to have identified very similar structure with no observable difference in the MDS plots or in the RMSE profiles (Figure 95b). This similarity is unsurprising, as all individual RFP images appear to show similar profiles. The 10-fold global MCT RE graphs (Figure 96) indicate the best splitting function is SSR, and all show a full MCT size of 3 splits (4 groups is optimal). From this the MCT is grown with SSR to three splits using the GPA consensus (Figure 97).

Figure 94: Horse colic analysis individual RFP MDS plots. The MDS plots are coloured by the predictor variable LESION.

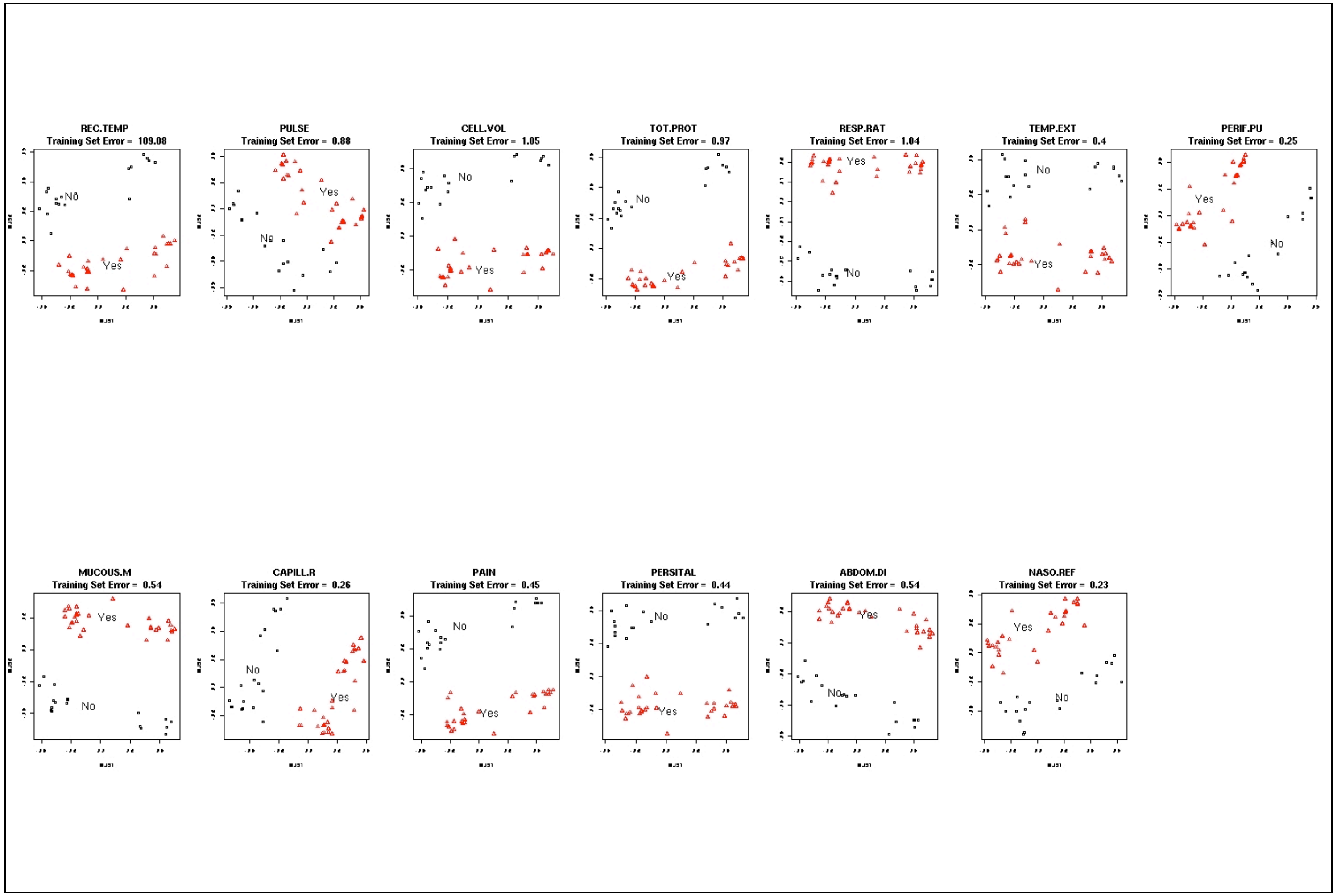
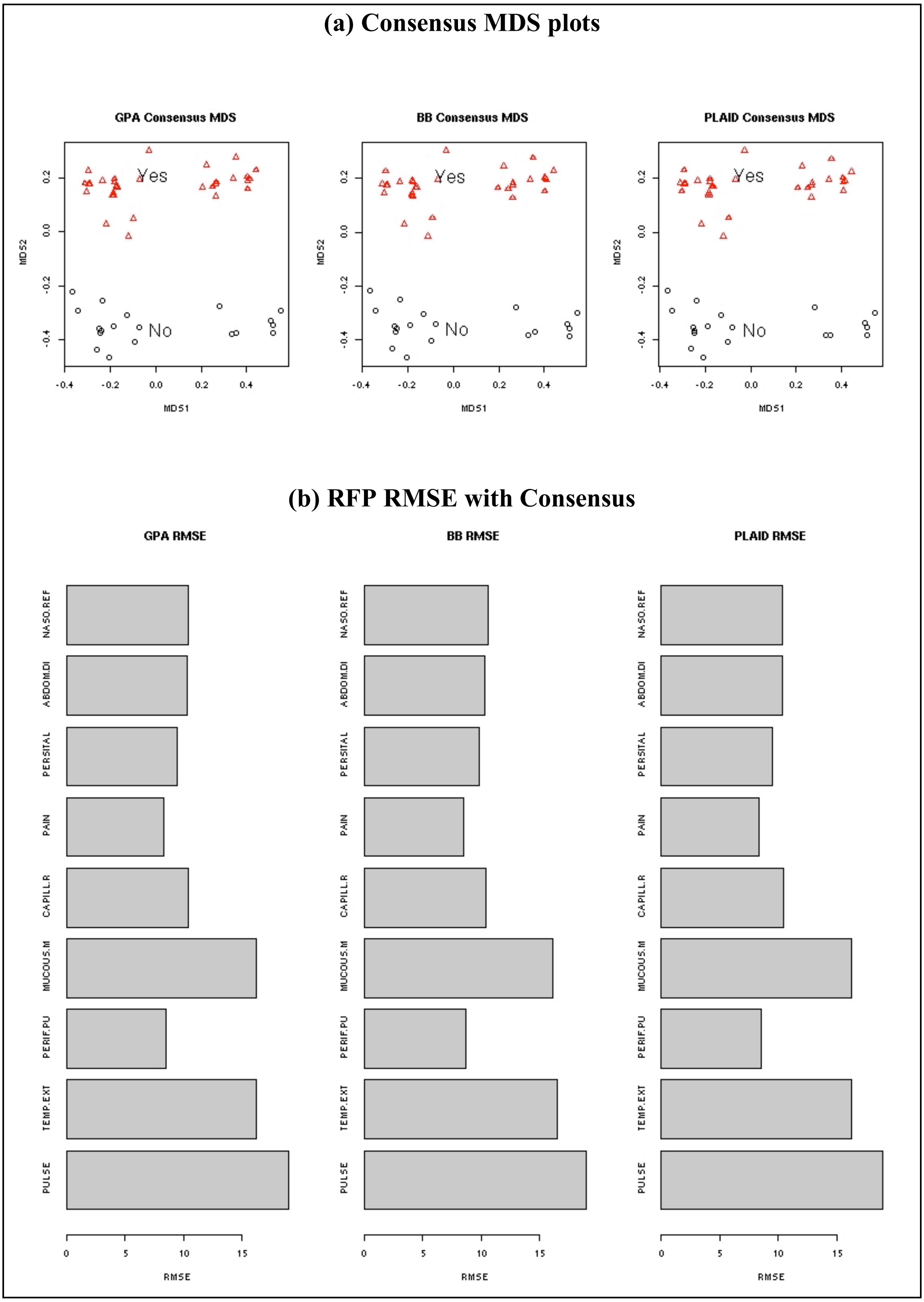


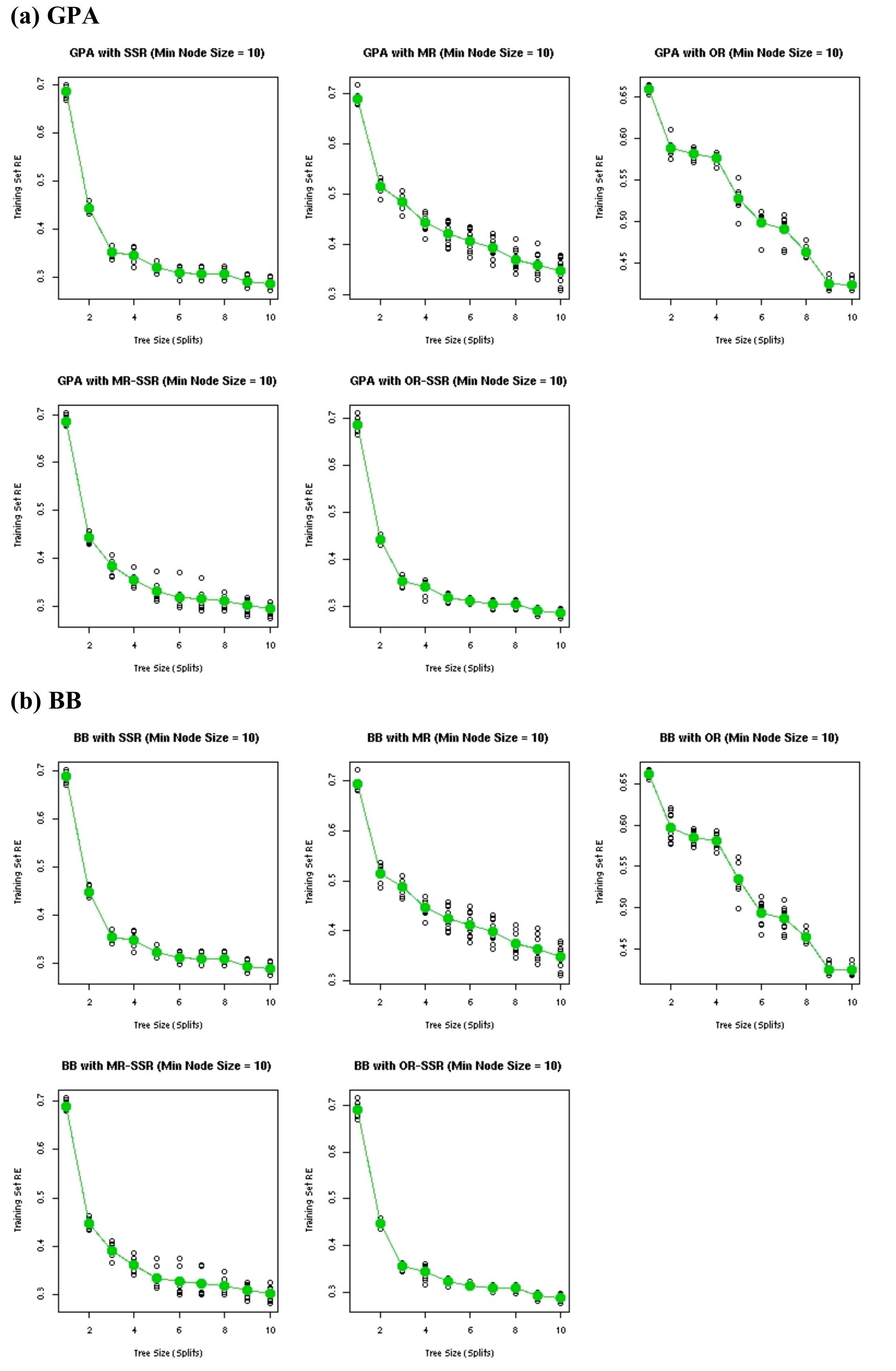
Figure 95: Horse colic analysis consensus MDS plots and consensus RMSE plots.



(a) Consensus MDS plots

(b) RFP RMSE with Consensus

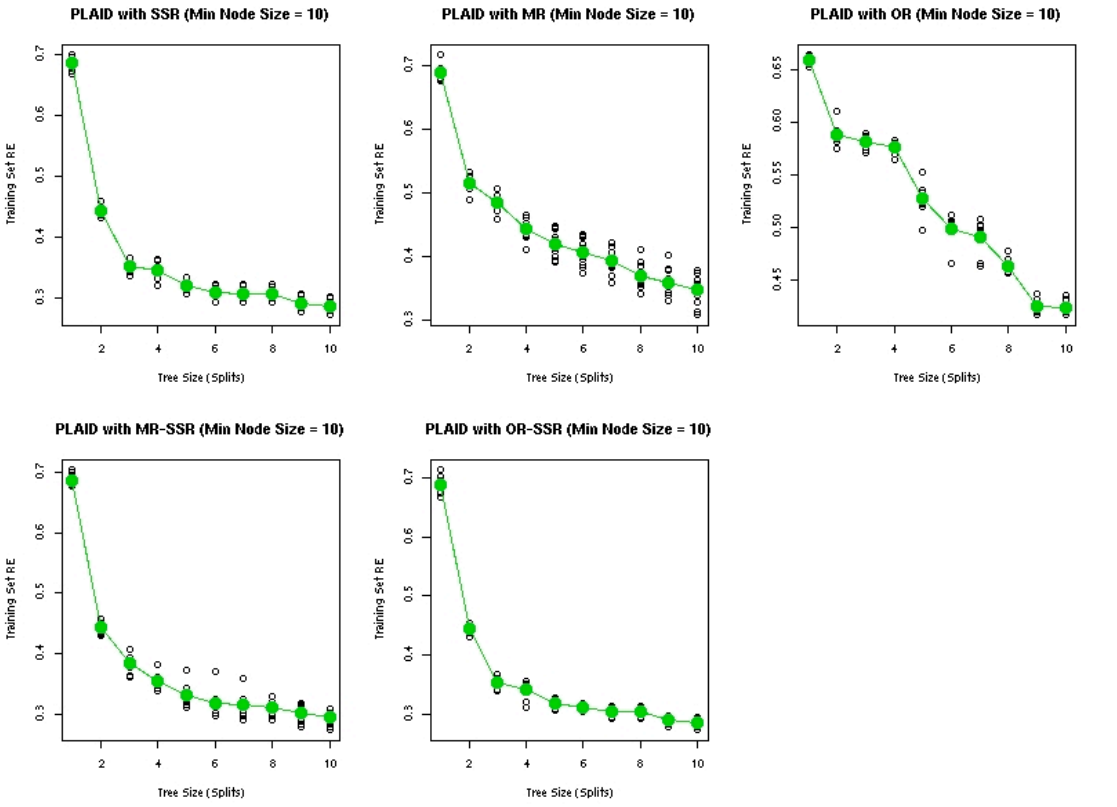
Figure 96: Horse colic analysis global MCT 10-fold CV RE curves.



(a) GPA

(b) BB

(c) PLAID



Taking into consideration the similarity in consensus, individual RFP configurations and the RE curves, it is not surprising that for each combination method with SSR splitting grown 3 splits, the same MCT is produced (Figure 97a). Interestingly, the least obvious group in the MDS plot, (Figure 97b, group 4) is the most well expressed in the MCT, showing a within node probability of 0.97. Also, each group, especially group 5, appears to be a combination of two groups which have not been identified. In fact these groups can never be fully resolved, even when the MCT is grown to 10 splits shown in Figure 106.

7.3.3.2 Complete response set global MCT plaid terminal node filtering

Plaid terminal node filtering takes the sub-matrices for each terminal node for each RFP and observes their structure. At a terminal node it is assumed that each RFP displays the same structure. The assumption is that each cell can be modelled sufficiently with the mean centroid of that sub-matrix. The PLAID consensus generation is seen as a way to test for the validity of this assumption. If the plaid model finds a κ_m of '1', it means that this RFP has a different count profile to the other RFPs. If the same structure is found the plaid consensus is the mean of all consensus matrices and each κ_m will be zero.

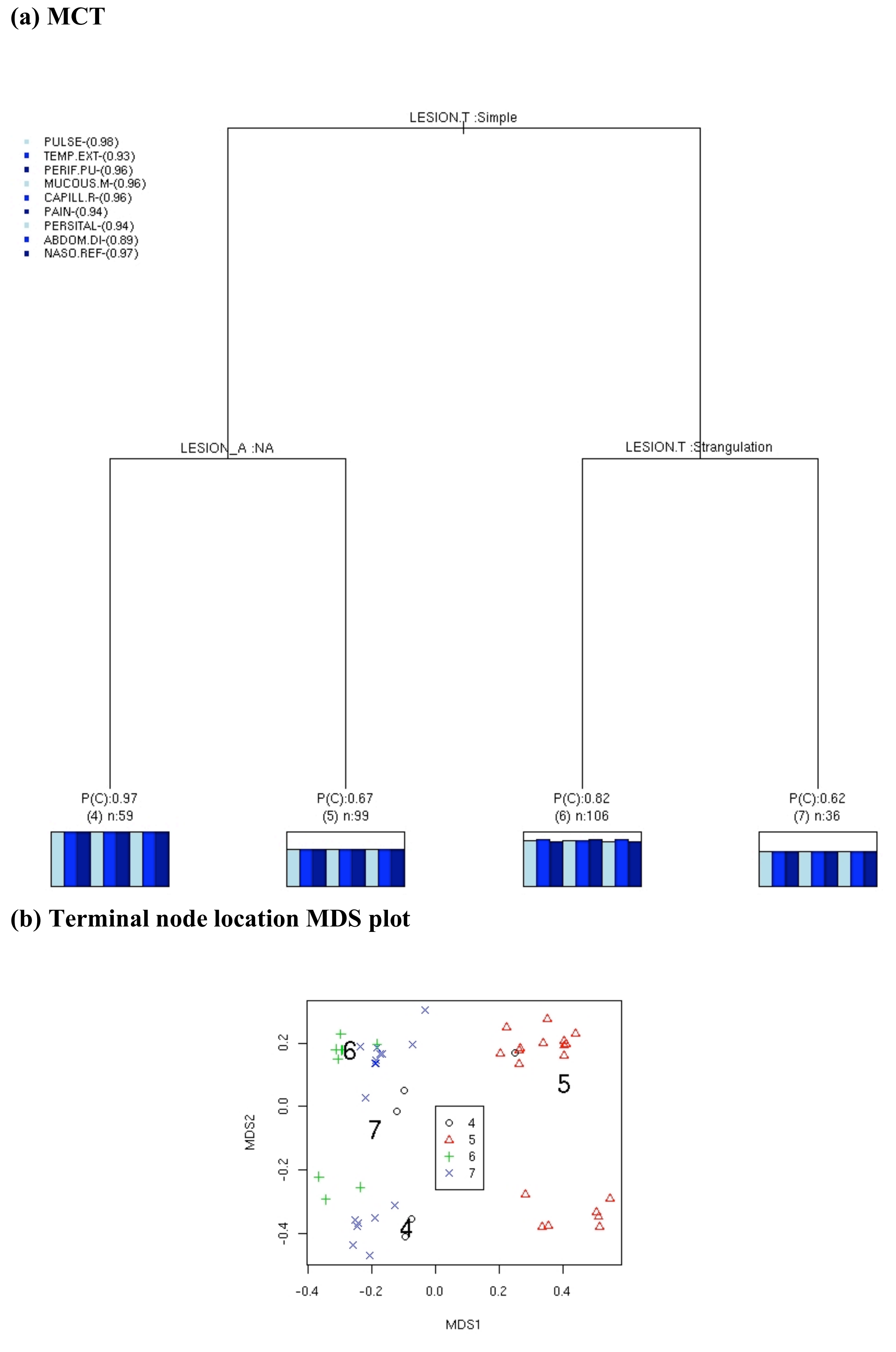
Running plaid terminal node filtering upon an MCT gives an indication of which RFPs express each group. The result of this process (Table 28) identifies variables that express that node's consensus structure as '0'. For those that deviate, the magnitude and direction of the deviation is estimated. The results clearly show that terminal node 4 is the most stable node with only CAPILL.R and ABDOM.DI expressing different configurations. Conversely terminal node 5 is the least stable with only PULSE, PERIF.PU, PERSITAL and ABDOM.DI expressing the consensus structure. Interestingly, terminal nodes 5 and 6 show the opposite expression structure, indicating a marked difference in profiles at these nodes. This fits with their relative positions within the tree.

Table 28: Horse colic analysis plaid terminal node plaid filtering results.

Response Variable	MCT Node 4	5	6	7
PULSE	0	0	4.57	0
TEMP.EXT	0	6.04	0	0
PERIF.PU	0	0	-6.92	-11.40
MUCOUS.M	0	5.03	0	0
CAPILL.R	17.03	-4.27	0	0
PAIN	0	6.99	0	0
PERISTAL	0	0	-3.57	0
ABDOM.DI	-17.29	0	5.79	-11.81
NASO.REF	0	-13.78	0	23.09

Terminal node filtering offers a means to test the homogeneity of each terminal node and investigate any variables that violate this assumption. However the MCT is built using information from all response variables, whether they are homogeneous with the MCT groups or not. It is possible that in a sufficiently complex response set that there will be sub-groupings of the variables that show different structure. We now propose an extension to the plaid combining method aimed at identifying these sub-groups before an MCT is build.

Figure 97: Horse colic analysis complete response set global MCT and terminal node location MDS plot.



(a) MCT

(b) Terminal node location MDS plot

7.3.3.3 Plaid response variable filtering algorithm

Plaid model RFP combination estimates a binary variable κ_m, which flags those RFPs whose configurations deviate from the mean configuration. An RFP with a κ_m of '1', has a different configuration from the mean, whereas a κ_m of '0' is considered to be adequately modelled by the background mean. Using a recursive algorithm described in Figure 98 it is possible to construct a search for similar configurations, by identifying those RFPs with κ_ms of '0'. The algorithm is stopped either when all κ_m s are either '1' or '0', or when the residual sums of squares between the RFPs and the combined configuration has converged.

If the residual sums of squares of the plaid model have converged, but there are still some κ_ms of '1', then plaid models considers these RFPs to be different but the effect of their difference is small. Therefore removing them does not improve the error in the modelled consensus structure. At this point, the RFPs are considered to be sufficiently homogeneous.

If the algorithm returns a subset where all κ_m s are found to be '1', it implies that all RFPs are sufficiently different from their background mean. Therefore no simple mean of the RFPs can be used to model the overall structure. If this occurs it is likely that PLAID combining will not yield the most accurate consensus matrix. In this case, a different combination method, designed to model heterogeneity between RFPs, such as the BB or GPA combination methods, should be employed to estimate the consensus.

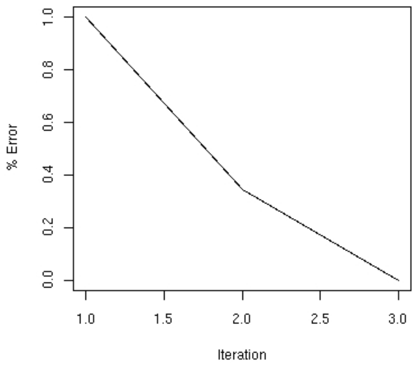
Figure 98: Plaid variable filtering algorithm.

```
1. Place all RFPs in subset A.
2. **While** subset A has RFPs within it **do**:
    a. Calculate the complete plaid model parameters for all RFPs in subset A as described in Section 3.5.3.
    b. Compute the plaid model error, $Q_i$, by (3.38).
    c. Compute the percent error relative to the error in the plaid model involving all RFPs in the initial subset A, $Q_0$.
    d. **If** all $\kappa_m = 0$ **then** stop, a good subset of RFPs has been found.
    e. **If** all $\kappa_m = 1$ **then** stop, the RFPs cannot be modelled well by a stable mean representation.
    f. **If** the percent error has converged but all $\kappa_m$'s do not equal 0 **then** stop, a reasonable subset of RFPs has been found.
    g. Update subset A with all RFPs with a $\kappa_m = 0$.
    h. Update subset B with all RFPs with a $\kappa_m = 1$.
3. Rerun the analysis on subset B.
```

The result of the plaid variable filtering algorithm in Figure 98is 4 groups of response variables shown in Table 29. For all variables but PERIF.PU the RMSE error between the RFP and the consensus configuration reduced, sometimes by over a half. Furthermore the variable groups found appear to make physical sense, with group 1 and 2 relating primarily to the horse's blood circulation function and group 3 relating to any observed pain the horse may be experiencing. Finally group 4 just contains MUCOUS.M (a variable describing the colour of the horse's eyes) and is grouped separately as it does not obviously relate to either heart function or observed pain.

The first group is found with some κ_ms not being zero. This is because the plaid model error is shown to be sufficiently small at two iterations (Figure 99). If the filtering algorithm is followed through to the third iteration, the first group of RFPs only contains TEMP.EXT. The results in Figure 99 show that at iteration 2 the RFPs contribute to less than 34 % of initial plaid model error at iteration 1. Because the error decrease from the second to the third iteration is small, the RFPs with a κ_m of '1' are shown to have a minimal effect on the homogeneity of the final consensus. Therefore they are considered sufficiently modelled by the consensus in the second iteration and the first group of RFPs is defined to be TEMP.EXT, PERIF.PU and CAPILL.R.

Figure 99: Horse colic analysis plaid variable reduction error convergence for the first group.



To investigate any improvements in resolution over these subgroups a global MCT is now built upon them. For the first group, the RE curves (Figure 101a) indicate a tree size of three or four splits is possible. This is one more split than the full response set MCT. In this MCT (Figure 101b) the first three splits are the same as the full MCT, and the additional split acts upon the full MCT's terminal node 5. The resulting terminal nodes 10 and 11, improve the probability of expression from 0.65 in the full MCT to approximately 0.77 and 0.72 respectively, in the reduced MCT. The MDS plot of the terminal node groups (Figure 101c) is more clearly resolved than in the full MCT with the noticeable difference in the group separation.

By the RE curve for the second group (Figure 102a), 3 splits are selected. The resulting MCT (Figure 102b) is identical to the full MCT in the splitting structure.

The only difference being in the MDS plot (Figure 102c), which appears to more clearly identify group 4.

The third group RE curve (Figure 103a) clearly indicates 2 splits, a smaller tree to the other groups. The splits made (Figure 103b) are the same as in the full MCT tree, however it does not make the partition to find nodes 7 and 8. The MDS plot (Figure 103c) shows a clear separation between nodes 3 and 5, however node 4 is not easily identified.

As the fourth group only has one variable, a consensus matrix does not need to be computed. The RE curves (Figure 104a) for this group indicate 4 splits as in the first group (Figure 104b). However by observation of the MDS plot (Figure 104c) the differences in the group structure between the two are apparent. Furthermore the fourth group more accurately defines groups 10 and 11. This is shown in the mean probability of expression within these terminal nodes increasing from 0.77 and 0.72 in the first variable set to 0.8 and 0.8 in the fourth set.

For each variable a strongly significant group profile is found (Figure 100). For categorical variables this was tested using a χ^2 test of independence between the group categories and the MCT terminal node labels, and for a continuous variable a one-way ANOVA was performed testing the mean difference between MCT terminal nodes. The significance of these profiles indicates the groups found by the MCTs are representative of structure within the responses. Correlation coefficients, r, are also presented to assess the strength of the relationships, for continuous variables Pearson's r is computed and for categorical variables Cramer's Phi is computed.

From the response variable profiles (Figure 100) it can be seen that the group structure found is weak. Group 1 appears to be defined by the temperature at the horse's extremities being either normal or reduced, a reduced pulse and a capillary refill time of less than 3 seconds. Group 2 finds MCT terminal nodes 4, 5 and 6 relating to no nasogastric reflux. The significant difference seen over the terminal nodes is driven by an elevation in pulse between nodes 4 and 7. In fact all of these profiles significantly highlight terminal node 4 as showing different structure. Terminal node 4 in these groups more often identifies the groups labelled 'normal' within the response variables. As a result an overall interpretation of these results is that RFP groups 1 and 2 are be primarily focused on identifying the profiles of a normal horse.

Group 3 has the strongest observed correlations however no clear group structure exists over the variables. Reversing the problem to a classification problem discriminating the groups in the predictor variable LESION, it is seen that only group 3 variables are used in the tree (Figure 105). This indicates the response variables within this group are those that are highly predictive of a single response variable LESION and therefore are determining the dominant grouping structure within the dataset.

Table 29: Horse colic analysis plaid response variable filtering results.

1. **Group one**

Variables:	TEMP.EXT	PERIF.PU	CAPILL.R
κ_m:	0	1	1
β_m:	0	-5.56	5.56
RMSE with filtered response consensus:	8.15	8.90	4.85
RMSE with full response set consensus:	16.26	8.49	10.41

2. **Second Group**

Variables:	PULSE	NASO.REF
κ_m**:**	0	0
β_m**:**	0	0
RMSE with filtered response consensus:	6.26	6.26
RMSE with full response set consensus:	18.90	10.37

3. **Third Group**

Variables:	PAIN	PERSITAL	ABDOM.DI
κ_m:	0	0	0
β_m:	0	0	0
RMSE with filtered response consensus:	4.59	6.58	6.49
RMSE with full response set consensus:	8.33	9.52	10.31

4. **Fourth Group**: MUCOUS.M

Figure 100: Horse colic analysis response variable group profiles.

(a) Group 1

TEMP.EXT

	MCT Node				
	4	**6**	**7**	**10**	**11**
Absent	2	17	3	5	0
Reduced	10	46	19	24	10
Increased	11	6	1	10	2
Normal	25	18	5	24	6

$\chi^2 = 43.044$
r = 0.19
P-Value = 0.003

PERIF.PU

	MCT Node				
	4	**6**	**7**	**10**	**11**
Absent	0	4	2	2	0
Reduced	3	49	19	23	9
Increased	3	1	0	1	0
Normal	39	25	8	34	9

$\chi^2 = 53.7028$
r = 0.21
P-Value = 0.00009

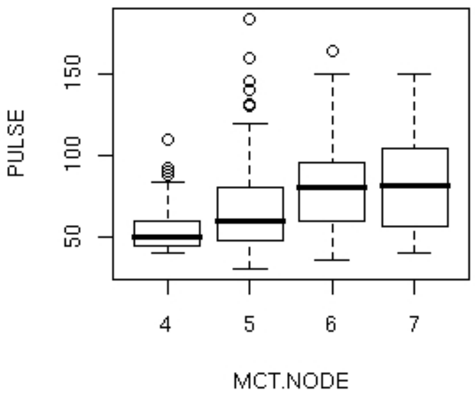
CAPILL.R

	MCT Node				
	4	**6**	**7**	**10**	**11**
>= 3 Seconds	0	41	15	14	5
< 3 Seconds	48	57	15	56	12

$\chi^2 = 30.0507$
r = 0.16
P-Value = 0.00009

(b) Group 2

PULSE



F = 36.908
r = 0.36
P-Value = $4.128 * 10^{-9}$

NASO.REF

	MCT Node			
	4	**5**	**6**	**7**
< 1 Litre	1	12	20	7
> 1 Litre	0	12	15	8
None	27	43	44	6

$\chi^2 = 25.6809$
r = 0.17
P-Value = 0.0005

(c) Group 3

PAIN

	MCT Node		
	3	4	5
Continuous severe pain	31	1	10
Intermittent severe pain	25	1	13
Intermittent mild pain	29	13	25
Depressed	31	10	18
Alert, no pain	2	23	13

$\chi^2 = 69.2889$
r = 0.34
P-Value = 0.00009

PERISTAL

	MCT Node		
	3	4	5
Abset	51	0	22
Hypomotile	56	23	49
Normal	4	6	6
Hypermotile	8	23	8

$\chi^2 = 65.4187$
r = 0.33
P-Value = 0.00009

ABDOM.DI

	MCT Node		
	3	4	5
Severe	23	0	15
Moderate	47	2	16
Slight	26	17	22
None	19	27	30

$\chi^2 = 50.2653$
r = 0.29
P-Value = 0.00009

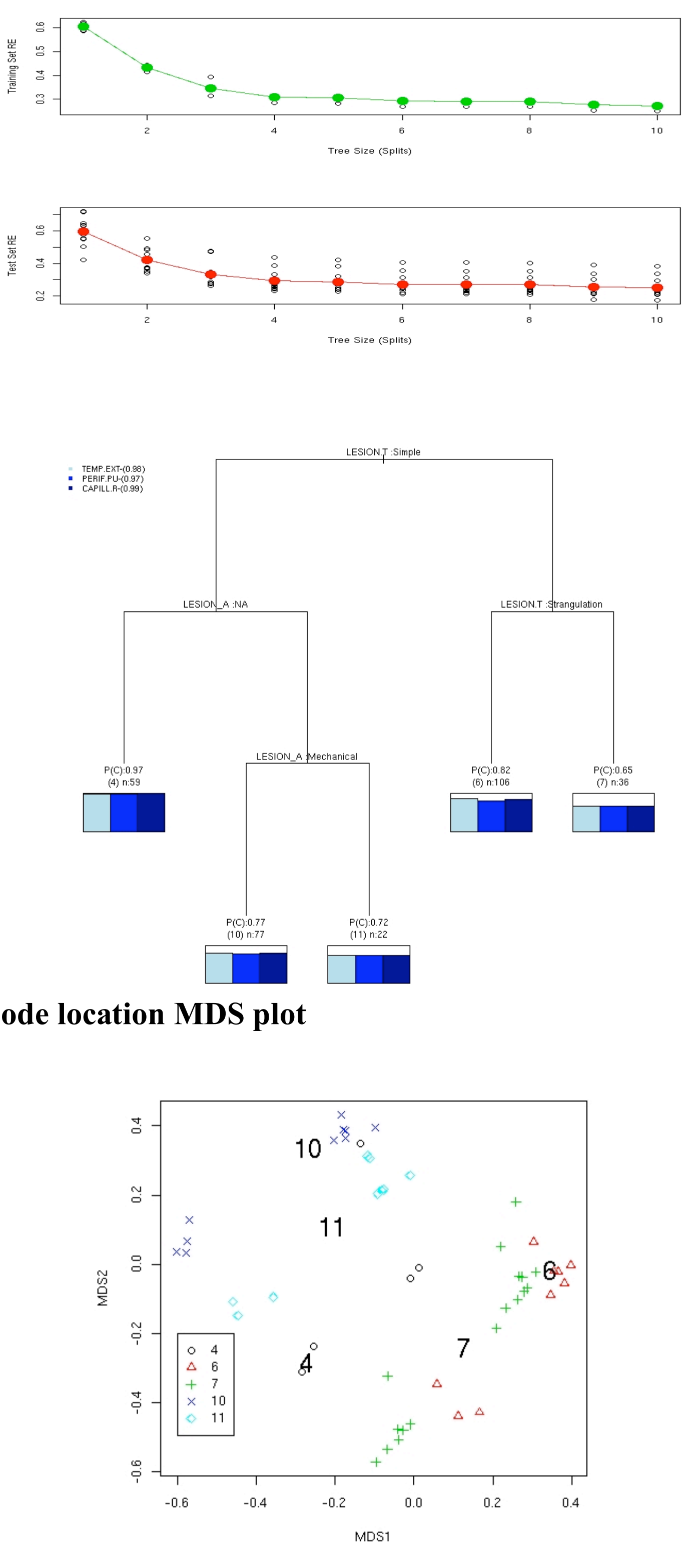
(d) Group 4

MUCOUS.M

	MCT Node				
	4	6	7	10	11
Dark Cyanotic	1	11	6	2	0
Bright Red/ Injected	2	6	10	5	2
Pale Cyanotic	1	28	4	8	0
Pale Pink	7	22	3	20	6
Bright Pink	10	10	2	5	3
Normal Pink	29	13	6	15	16

$\chi^2 = 95.29$
r = 0.28
P-Value = 0.00009

Figure 101: Horse colic analysis plaid filtered variable group one MCT results.

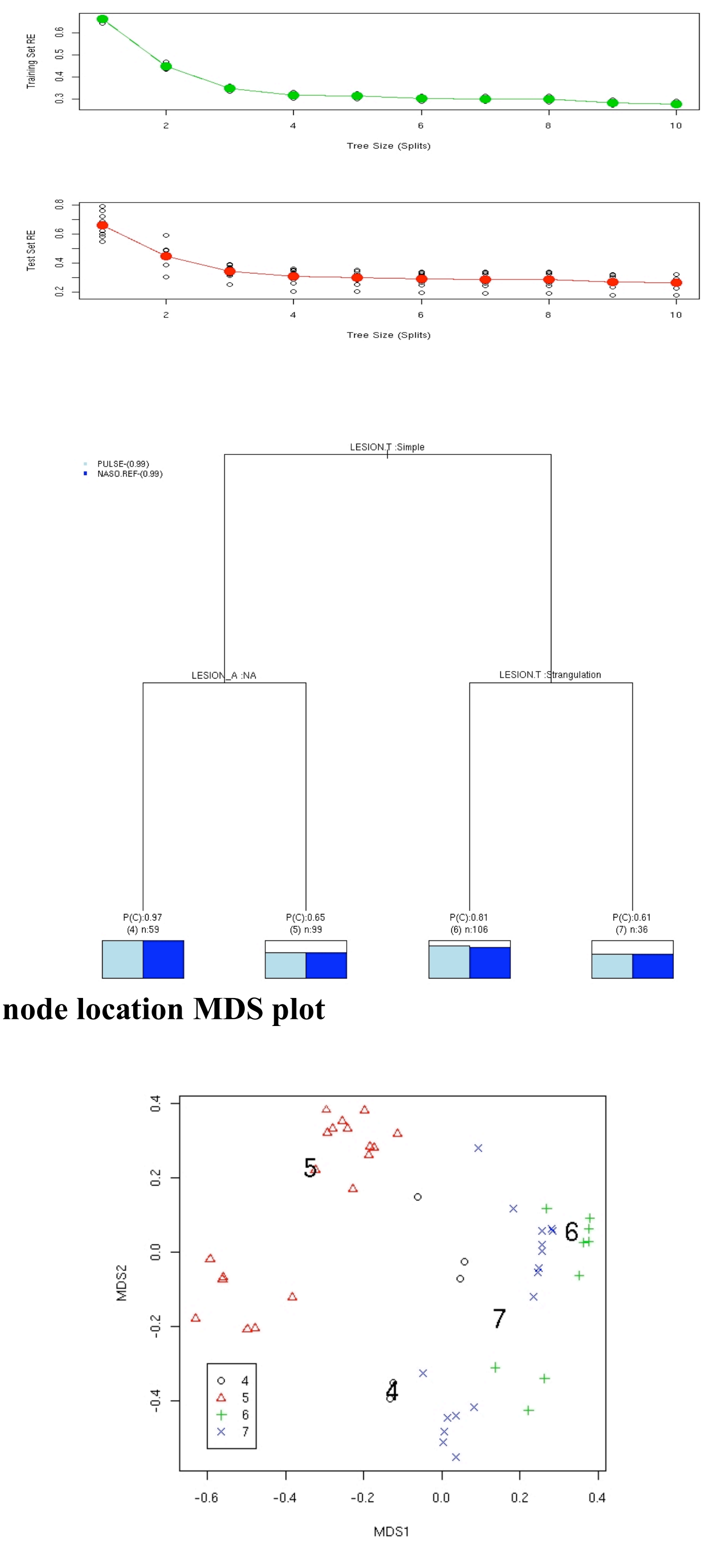


(a) 10 fold CV RE curve

(b) MCT

(c) Terminal node location MDS plot

Figure 102: Horse colic analysis plaid filtered variable group two MCT results.

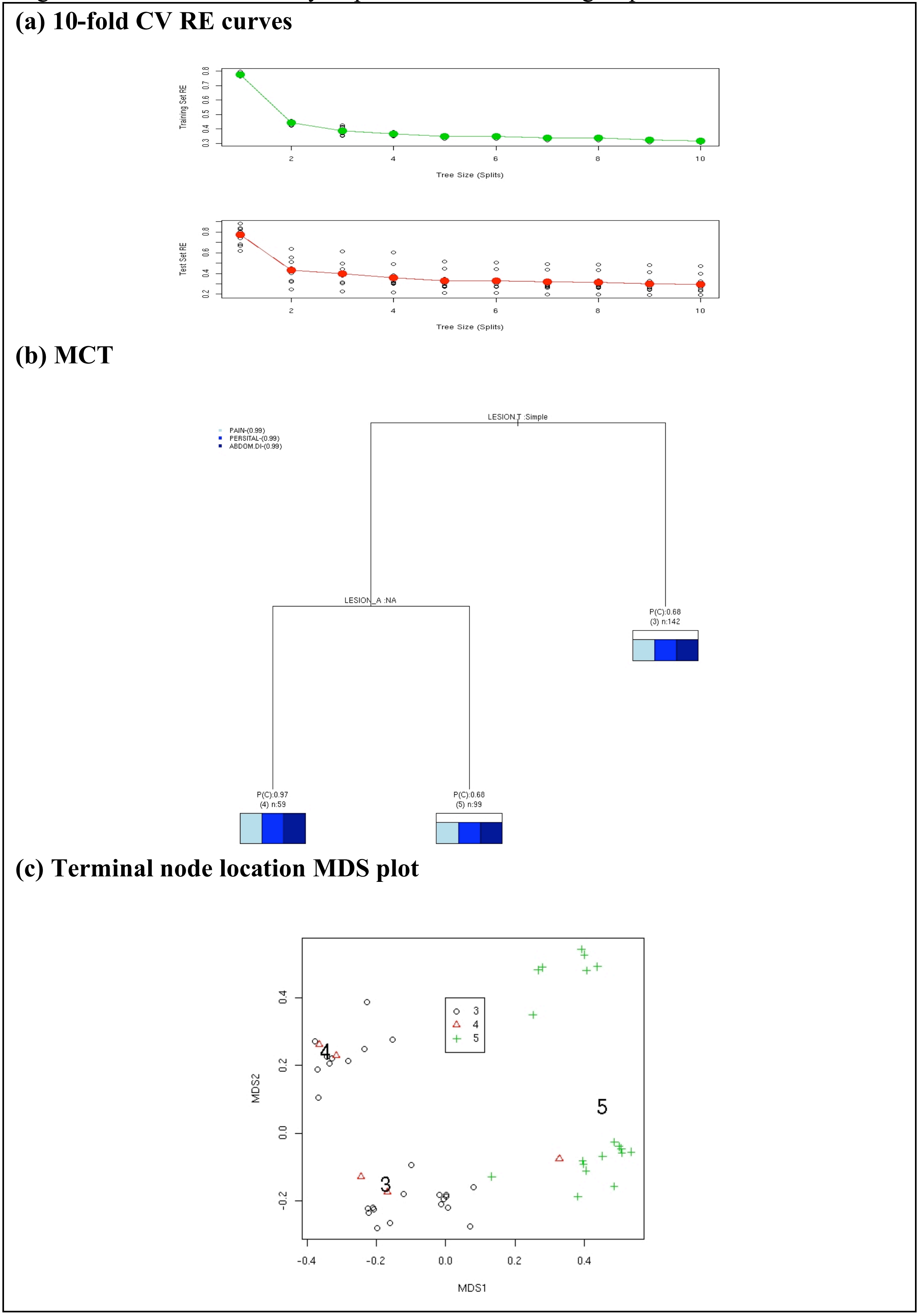


(a) 10-fold CV RE curve

(b) MCT

(c) Terminal node location MDS plot

Figure 103: Horse colic analysis plaid filtered variable group three MCT results.

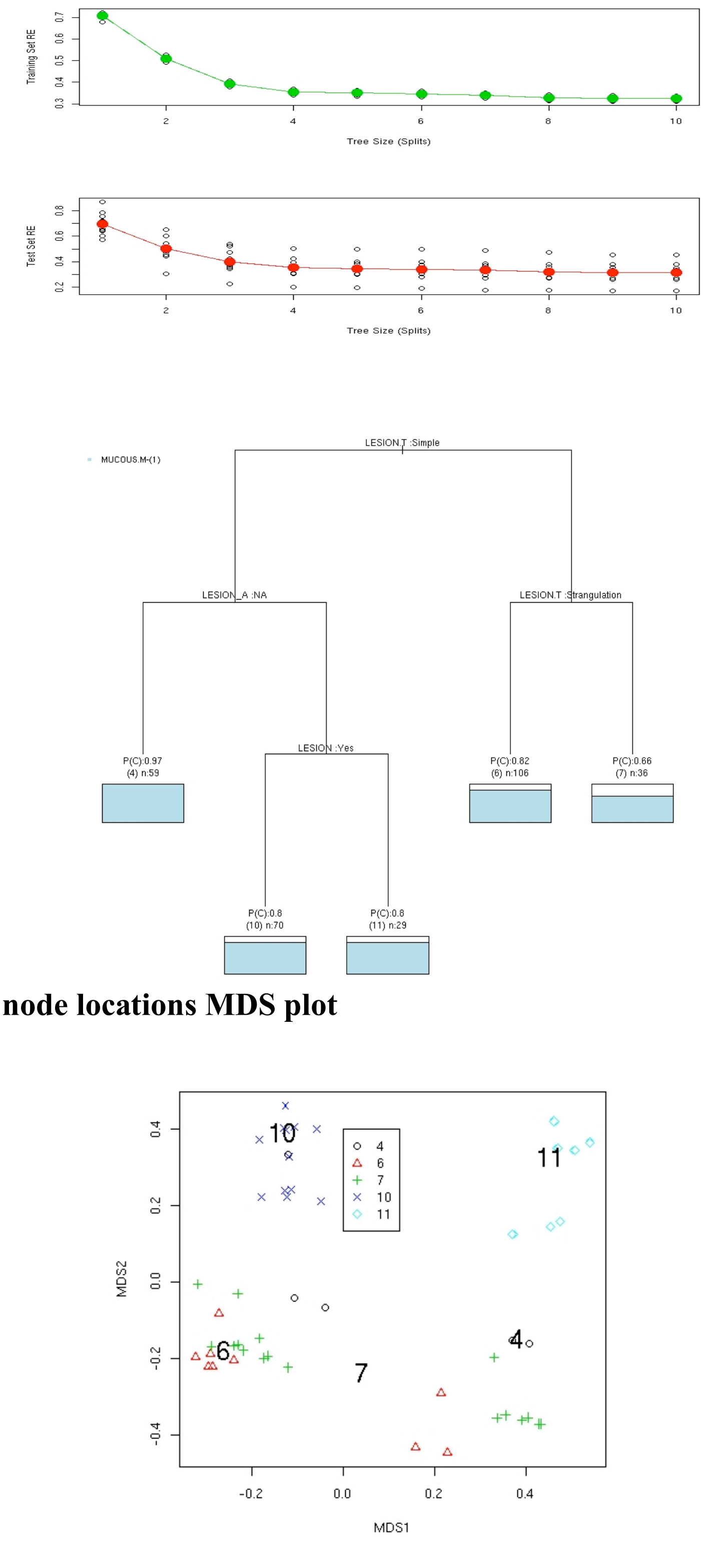


(a) 10-fold CV RE curves

(b) MCT

(c) Terminal node location MDS plot

Figure 104: Horse colic analysis plaid filtered variable group four MCT results.



(a) 10-fold CV RE curve

(b) MCT

(c) Terminal node locations MDS plot

Figure 105: Horse colic analysis classification tree classifying the groups within predictor LESION by the entire response set. (Correct classification rate of 77.33 %).

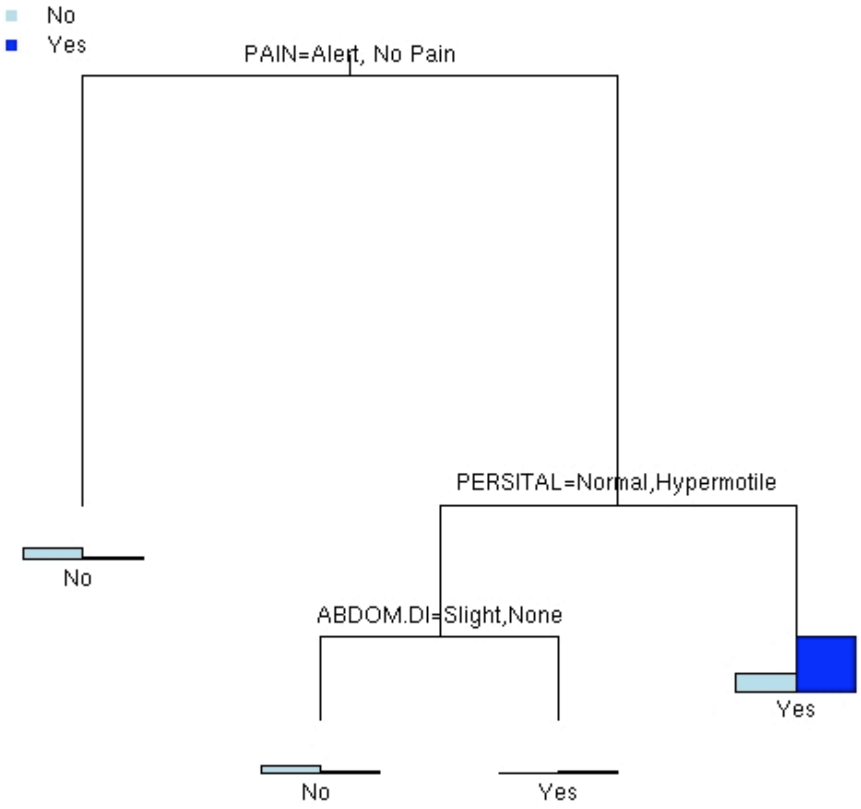
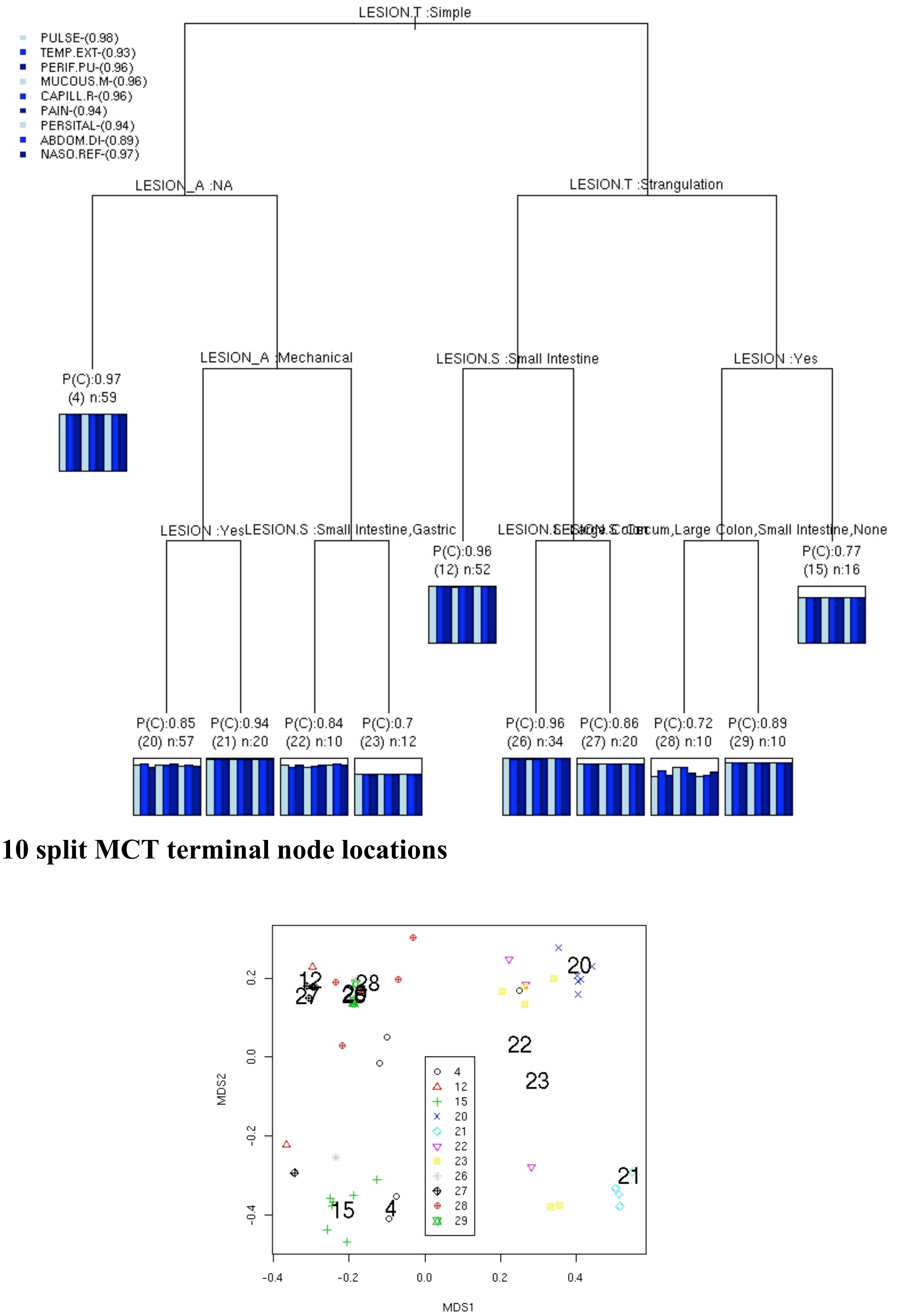


Figure 106: Horse colic analysis MCT grown to 10 splits.

(a) 10 split MCT



(b) 10 split MCT terminal node locations

7.4 Horse Colic Summary

The horse colic dataset is a good example of a profiling style analysis with MCTs. Firstly, using the a simple MRT on either the Gower distance matrix or binary substituted representation of the profiling set resulted in poor results. The improvement gained from moving to the random forest and treeboost proximity matrix is considerable. Groups that are common to both predictor and profiling sets now become obvious and easily found using the MCT splitting criteria. These methods give good indications of the structure to be found within the analysis however provide little detailed information on the composition of the groups.

Using MCTs it is possible to observe the grouping structure of each individual response variable and the relationship with the consensus matrix. It allows for the terminal nodes of the resulting MCT to be simplified using plaid filtering. By using MCTs with plaid terminal node filtering a two-way clustering is performed, where within a terminal node lie a subset of response variables and observations that define the common profile within the group.

A pre-processing step can be taken with the recursive filtering algorithm, allowing for an initial clustering of the responses based on the structure within their RFPs. This analysis highlights the complexities within profiling studies, as each response group displays a different subset of groups. What is interesting in this analysis is not the differences but the similarities between the subsets: in this case the splitting variables used in the tree and the tree size. It is clear that modelling a subset of variables produces a more accurate result, however this improved accuracy relates to the same groups found in the overall analysis. It did not change the consensus structure completely, but reinforced the groups found in the overall consensus.

A major issue with the horse colic analysis was the high level of missing values. The results for the plaid filtering are dependant on the original global MCT model shown in Figure 97 and therefore if any bias in the missing values exists it will be obvious in this model. The percent of missing values in each terminal is shown in Table 30 show that terminal nodes 5 and 6 contain the 69 % of missing values and 4 and 7 only contain 31 %. However comparing this distribution to the relative size of each terminal node it is seen that the missing values are distributed with terminal node size. Therefore no obvious bias towards any particular group of missing values is observed.

Table 30: Horse colic analysis global MCT terminal missing value distribution.

MCT Node	**4**	**5**	**6**	**7**
% Missing values	0.19	0.34	0.35	0.12
Relative terminal node size	0.22	0.33	0.35	0.12

8. Discussion

The aim of this thesis is to extend tree based methods to handle a mixed type multivariate response. To do this a series of methods have been developed. Firstly, mixed type extensions to a multivariate tree are implemented by transforming the response using either the Gower distance, or binary substitution. These techniques offer a simple solution to a complex problem, but provide little in the way of understanding the result. Secondly, to improve on the performance of a single tree, multivariate tree based ensemble methods are also developed. Ensemble methods improve the predictions on the multivariate responses, and by binary substitution, are further extended to mixed type response sets. Multivariate tree-based ensembles are shown in this thesis to be powerful methods for profiling.

One key feature provided by tree-based ensemble methods is their proximity matrices. These proximity matrices are identical to consensus matrices that can be produced over a cluster ensemble. This changes the interpretation of a tree-based ensemble to that of a consensus clustering algorithm. A result of this interpretation is that the predictive performance of the ensemble becomes a key statistic in determining the quality of the final clustering solution. By using the ensemble predictive performance the problems in determining the accuracy and reproducibility of a cluster ensemble are reduced.

The major contribution of this thesis is the development of multivariate consensus trees (MCT) for mixed type clustering or profiling. MCTs combine ensemble proximities into one overall consensus matrix in an analogous step to the cluster ensemble search for the overall partition. This provides more information on the accuracy of the final solution with the ability to analyse the individual group structure of each response variable in the analysis. MCTs partition the consensus matrix to find the optimal partition. This procedure uses decision rules to map the predictor variables back over the consensus to allow for an understanding of the origin of the final groups in the optimal partition. These rules also make MCTs a predictive clustering or profiling algorithm allowing them to easily group new observations without altering the original model. This predictive ability allows MCTs to cross-validate estimates on the number of groups and overall group accuracy.

Before opting for the more complex and computationally expensive solution as implemented in MCTs, using the simple tree and ensemble methods can be useful. The Gower distance metric and binary substitution transformation of the response set are common ways of finding groups in mixed type domains. In this thesis the results of these approaches are remarkably similar to each other as they both assume a Euclidean relationship between categorical and quantitative variables. This similarity is highlighted in the breast cancer dataset analysis. In this example both approaches grow the same tree and the MDS plots show very similar group structure (Figure 65, Figure 67). Binary substitution is the more flexible of the two approaches as it can also be used with ensemble tree methods. In the case of obvious structure these simple extensions will work.

A major problem for binary substituting of the response is that of dimensionality. Binary substitution inflates the number of variables within the response by the total number of levels within each categorical variable. In the case of the breast cancer analysis the 9 original categorical variables were transformed into 89 binary variables. Although it has been shown that multivariate tree based methods can handle a large response set (Smyth, Coomans and Everingham 2006), the understanding of the final result is impaired by the dimensionality. Furthermore as the response set is treated as a whole, filtering out unimportant variation is not possible.

Multivariate extensions to tree based ensembles are shown to clearly improve group resolution within the MDS plots of the proximity matrices. On comparison between multivariate random forest and treeboost notable differences in the group structure of the responses are observed. The group structure within the random forest proximity matrices more closely matches that observed using the Gower distance and binary substitution. However treeboost appears to find a consistently different group structure as seen in the thyroid and breast cancer analyses. This difference does not manifest itself in performance, with the final grouping of the treeboost proximity matrix outperforming the final random forest proximity.

Despite the improved accuracy observed when determining the groups over the treeboost proximities, they are not appropriate inputs for the MCT consensus construction. There are two reasons for this:

1) Boosting models are sensitive to the shrinkage parameter. This prohibits automated running of the model as required for MCT construction. The action of the shrinkage parameter means that simply increasing the number of trees within the model will not achieve optimal performance (Hastie, et al. 2001). Random forests however can be easily tuned by increasing the number of trees within the forest to achieve optimal performance, and because of this are ideal candidates for MCT construction.

2) The trees in a boosting model are dependant upon each other. This means that each tree does not contribute equally to the construction of the proximity, a fact that is not reflected within the proximity matrix itself. This violates the assumption of a binomial distribution of the counts and could seriously affect the combination methods.

The analysis of the consensus matrix with the MDS plots must go hand in hand with a heat map of the reordered matrix. The structure of the groups with the MDS plots does not represent their structure within the dataset but how well that group has been predicted by the ensemble. The result of this is that groups that are poorly predicted will be large and noisy within the MDS plot. These groups will also have a relatively low probability of expression.

From the base tree and ensemble methods it is clear that trees are highly suited to mixed type clustering and profiling. The primary feature of tree-based methods is the ensemble proximity matrices. By partitioning these matrices it is possible to simultaneously view a logical decision path that predicts each group in the form of a tree and the relationships between these groups within the MDS plots, a feature that is not available with any other unsupervised technique. This allows for a detailed understanding on how the groups within the predictor set match the response set. However as the response set is treated as a whole, they do not allow for clear understanding of how well each response variable expresses each group. To do this the more individualistic analysis of MCTs is required.

MCTs are designed for simultaneous analysis of relationships in both the response variables and between the response and predictor variables. This is done by individually analysing the group structure within each response variable. By combining these structures MCTs not only retain all the functionality of the random forest proximity matrices but also can improve on the group resolution. In fact this thesis shows the performance of MCTs for unsupervised classification can be comparable to the performance of a classification tree. Also by analysis of the individual RFP structures it is possible to filter noise and unrelated variables from the response set by using both performance diagnostics and plaid combining.

By extending PLAID combining, MCTs offer an algorithm to filter the response variable set. Plaid filtering is implemented in two ways, firstly to cluster the response variables before construction of an MCT, and secondly to test the assumption of homogeneity within the terminal nodes of an existing tree. Plaid filtering extends MCTs to be a two-way technique, where a group is defined both on a subset of observations and variables. In the horse colic analysis plaid filtering is used to cluster the variables within the response variable set. Over the four response variable subgroups found, different group structure within them is observed. Furthermore, the consensus produced from each subgroup is a more accurate consensus in terms of RMSE between RFPs of the subgroup and the overall model consensus.

The horse colic results give a clear indication of the power of plaid filtering. Firstly the algorithm removes the dominant variation corresponding to the normal symptoms of a horse, and then places together the variables related to the lesion groups. In addition the profiles found for each subgroup are different and also strongly significant. However the relationships in terms of correlation observed over the terminal nodes are weak ranging between 0.16 to 0.36. Therefore plaid filtering is shown to be effective even when the observed grouping structure is weak.

However, as plaid filtering is finding groups over RFPs generated from a random procedure, care should be taken to ensure that all structure in these matrices has been fully resolved. This can be done easily by increasing the number of trees within each forest and observing the structure change. If the RFPs themselves are unstable, then plaid filtering will also be unstable.

Much of the effort in this thesis is spent of testing the effect of various parameter specifications in the three stages of MCT construction (Table 2). The estimation of the overall consensus matrix from the individual RFPs is the first major complexity within the MCT algorithm. Three combination methods are proposed in this thesis, GPA, BB and PLAID. Both GPA and PLAID define the overall consensus by minimising the square error loss between each individual RFP and the consensus. BB does not minimise a loss function but rather provides a robust estimation of each count within the proximities by estimating their overall probability distribution. As a result it is expected that different combination methods will provide a different consensus solution.

By analysing the RMSE errors between the RFPs and the consensus matrix it is possible to assess the quality of the combination. This provides a response variable importance statistic for the overall MCT. Strong similarity is found in the resulting consensus matrices from each combination method. From the results it appears that GPA and BB are finding very similar structure as they produce the same global and local MCT in the breast cancer analysis (Figure 79, Figure 81) and show the same performance convergence in the sensitivity analysis (Table 16, Table 17). PLAID combining however produces a different global MCT for the breast cancer dataset (Figure 79), shows a different performance convergence in sensitivity analysis (Figure 79, Figure 81) and shows a much increased RMSE for the 10 uneven but clear group simulation tests (Figure 39). However whether these differences translate into reduced clustering accuracy is not clear. In the sensitivity analysis using PLAID combining shows a less accurate consensus that resulted in a reduced performance of the overall tree. However in the breast cancer dataset, using PLAID combining results in the most accurate tree.

The inconsistent performance of PLAID combining is most likely due to the plaid model's search for common structure over the RFPs. In this thesis the plaid model is only run to a single layer. This may result in smaller groups being modelled in later layers, as the common structure in the first layer is likely to favour the larger groups. This is what is observed in the sensitivity analysis. With plaid models the MCTs grown using the PLAID consensus do not finding the smallest group (group 3) and in the ten group simulation experiment with large and small group sizes the RMSE for plaid combining is obviously the greatest. These results show that PLAID combining may not resolve smaller group structure over the RFPs.

An obvious solution to this problem is running plaid combining to more than one layer. However, as different group configurations will exist within each additional layer any interpretation of what variables contribute to the groups in the final consensus will be confused. This removes one of the most important features of the plaid combining algorithm. Another approach is to change the background layer from an average consensus matrix to one produced by BB or GPA. This approach biases plaid models to the structure found by BB and GPA. A flow on affect is to change the interpretation of the plaid parameters. Instead of modelling the deviations from the mean consensus they are modelling the deviations from a modelled consensus. As this modelled consensus has no simple expression, the interpretation of the plaid parameters as estimating deviations from homogeneity does not hold.

This thesis also assessed the effect noise variables within the RFPs will have on the final consensus solution. The results showed that the consensus generation procedure of MCTs was found to be remarkable resistant to added noise within the response set. In these experiments it is shown that the consensus configuration has the same group structure as the original despite the addition of pure randomness within the consensus generation procedure. Furthermore the RE curves accurately estimate the number of clusters, and the accuracy of the resulting partitions is found to be comparable or exceed that of K-means.

Once an overall consensus has been estimated the task is now to partition the matrix to find the groups. To do this five splitting criteria are developed. These criteria search over all decisions within each variable in the predictor for blocks of observations with high similarity within the consensus matrix. In an ideal case the decisions found will reorder the consensus matrix into a block diagonal structure where the similarities on the block diagonal are high and the similarities within the off diagonal blocks are low. Of the five splitting criteria developed, one observes the group variance structure (SSR), two utilise the count structure of the cells within the matrix (MR and OR) and the last two are combinations of MR and OR with SSR.

The quality of each splitting criteria is best assessed by observation of their respective RE curves. Over the simulation tests these curves were produced for each criteria for each experiment. Overall it appears that MR and MR-SSR produced curves that are less accurate than the other splitting criteria (Figure 35, Figure 41, Figure 45). This is shown by consistent high variability within the cross-validated performances. The other splitting methods performed indistinguishably as the RE curves are closely matched and the misclassification performances similar. However when moved from the sterile domain of the simulation experiments to an actual dataset a clear interaction between the performance of the splitting criteria, combination method and random forest parameters emerged.

Considerable effort has been made in this thesis to quantify the interaction between the splitting criteria, combination method and random forest parameters for both global and local MCTs. Global MCTs have a clear interaction with random forest tree terminal node size. The structure of this is that if the terminal node size is set too small by increasing the tree size optimal performance can be reached (Table 16). This interaction is seen to be mostly independent of combination method. However local MCTs seemed only to be sensitive to terminal node size (Table 17). Here the terminal node size must be specified as close as possible to the smallest group size in the data.

These results are not surprising given how the two types of MCTs are grown. As local MCTs recompute the consensus matrix it is difficult to optimise the performance of the base random forests. Furthermore the choice of MCT splitting criteria is a sensitive parameter as the accuracy of the intermediate consensus matrices is dependant on the previous decisions within the tree. Global MCTs do not suffer as severely from this problem because the response is constant and therefore it is easier to optimise the important parameters. These differences are highlighted in the sensitivity analysis of the Vietnam data.

Local MCTs, when optimised, show a more improved resolution of the groups. This improvement highlights the power of a localised clustering solution. The resolution is improved as once obvious groups are removed from the analysis, more attention can be paid to separating the groups that are closer together. This is strongly highlighted in the breast cancer dataset analysis. To get the same performance the local MCTs require 3 splits whereas global MCTs require 5. This increased split accuracy is highlighted in the identification of the benign group. Global MCTs identify the majority of the benign group at terminal node 15, and much of the work in the early splits of the tree is dedicated to shaving off smaller malignant sub-groups (Figure 79). However local MCTs find the majority of the benign group first, in terminal node 3, and then use the other two splits to find the less obvious malignant sub-groups (Figure 81). This implies that local MCTs will find the most obvious groups first, whereas global MCTs are likely to favour smaller groups.

MCTs however are limited by their tree structure when finding groups. In the thyroid, Vietnam and breast cancer analyses it was found the performance of MCTs approaches the performance of a classification tree. However in the thyroid analysis it was apparent that K-Means and PAM on the consensus matrix found by MCTs identified the known groups more accurately. For example, the local MCT for the thyroid dataset misclassifies 15 observations, whereas on the ACM, K-means misclassifies 12 and PAM 10 observations. The reason for this is that tree-based clustering methods are bound by groups that are separable by a single decision on a single predictor variable, where as K-Means and PAM are not. A possible solution to this is to define a multivariate or linear combination of splitting functions (Breiman, et al. 1984, Brodley and Utgoff 1995).

MCTs when run correctly are a powerful technique for clustering or profiling. However there are a lot of parameters that can serious affect the accuracy of the final solution. For a reasonable dataset as computation time for MCTs is considerable a course of action to determine a reasonable set of parameters for a MCTs analysis is now described:

1. The first step is to produce a single multivariate tree upon the dataset. If mixed types exist then use binary substitution or the Gower distance approaches. From this analysis it is hoped that the following information is gained:
    a. To determine appropriate tree and terminal node sizes for the ensemble methods. These can be determined through observation of the RE graphs.
    b. To assess the predictive performance of that tree. If there is no stable tree observed from the RE graphs then it is unlikely that this will change with any further analysis.
2. Once you have appropriate estimates for tree and terminal node size the next step is to build a simple multivariate random forest or treeboost model using these

parameters. From these models the most important piece of information is the error convergence plots. It is strongly recommended that a reasonable independent test set be used to assess the performance of the ensemble. If the predictions upon this test set do not converge then no stable trees can be built, and as a result this analysis will not find stable groups. If the performance does stabilise it is recommended that more trees well past the point of convergence be added to the model to ensure that this stability remains.

3. Once the ensemble methods are stable then the MCT approaches can be considered. Firstly observe the structure within the plots of the proximity matrix to get an idea of the quality of the group structure. Then produce the RE curves to partition this proximity matrix for each splitting criteria. If a stable group structure has been found the elbow should appear at the number of groups observed in the proximity matrix plots. If this is the case then MCT methods are likely to find a representative set of clusters.
4. The decision to go to the local or global MCT methods using a combined consensus matrix should be determined on the number of response variables available. If there are many responses then filtering out some before combining is recommended. In the horse colic analysis this was done on the basis of the performance of the random forest for each response variable. From here it is advisable to perform all combination techniques but only growing a global MCT on each. Once the global MCT parameters have been optimised, then a local MCT approach using similar parameters may be attempted. It should be noted that local MCTs are remarkably more sensitive than global MCTs to the choice of parameters and may take some time to optimise.

5. After the MCTs are grown and the results make sense, plaid filtering can be performed. However it is advisable that a stable model be found before performing this step as reproducible proximity matrices for each varaible are required.

It is hoped that this guide will produce stable MCT solutions, however the final result will be dataset dependent.

9. Conclusions

This thesis has developed models for mixed type clustering or profiling. The core idea within this thesis is that groups found to describe a dataset must be predictive of each variable within it. The methods developed in this thesis use tree-based ensemble techniques to predict the data, and cluster ensemble ideas to identify the overall grouping structure. This combination of ideas culminated in the development of a new algorithm called Multivariate Consensus Trees (MCT).

Multivariate Consensus Trees, in this thesis have been shown to find more accurate grouping structure than either hierarchical agglomeration, K-Means or PAM. Furthermore they enable an analysis of the found groups in terms of: "*which predictor variables determine the groups?*"; "*which response variables express these groups?*" and the probability that these groups are representative of the data. MCTs also allow for pre and post-processing steps, using plaid models, to filter out response variables that do not express the groups found by the MCT. These features of MCTs make them a unique tool for finding and understanding the grouping structure over a mixed type dataset.

The focus of MCTs is in finding groups on a multivariate mixed type response. However this thesis has also suggested methods for mixed type prediction using multivariate extensions to tree-based ensembles using binary substitution of categorical variables within the response dataset. Multivariate random forests and treeboost are new methods of predictive profiling analysis that can highlight grouping structure, but are more focused on creating an accurate predictive model. Tree based models are resistant to overfitting problems and can handle large datasets. These features are highly desirable for any multivariate model.

Overall this thesis has exploited the flexibility of trees in handling mixed data types and extended them to a predictive multivariate ensemble. Moving from prediction to clustering this thesis views a tree-based ensemble as a consensus clustering algorithm. The result is multivariate consensus trees, a tree based clustering and profiling tool for mixed data types.

10. References

Ana L. N. Fred, and Anil K. Jain. (2005), "Combining multiple clusterings using evidence accumulation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27, 835-850.

Ben-Hur, A., Elisseeff, A., and Guyon, I. (2002), "A stability based method for discovering structure in clustered data," *Pacific Symposium on Biocomputing*, 6-17.

Breiman, Friedman, Olshen, and Stone (1984), *Classification and Regression Trees*, Chapman and Hall.

Breiman, L. (1996a), "Bagging Predictors," *Machine Learning*, 26, 123-140.

Breiman, L. (2001), "Random Forests," *Machine Learning*, 45, 5-32.

Brodley, C. E., and Utgoff, P. E. (1995), "Multivariate Decision Trees," *Machine Learning*, 19, 45-77.

Burnham, K. P., and Anderson, D. R. (2002), *Model Selection and Multimodel Inference. A Practical Information-Theoretic Approach* (Second ed.), Springer-Verlag.

Buuren, S. V., and Heiser, W. J. (1989), "Clustering N objects into K groups under optimal scaling of variables," *Psychometrika*, 54, 699-706.

Carroll, J. D., and Chang, J. J. (1970), "Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition," *Psychometrika*, 35, 283-320.

Coomans, D., Broeckaert, M., Jonckheer, M., and Massart, D. L. (1983), "Comparison of Multivariate Disciminant Techniques for Clinical Data - Application to the Thyroid Functional State," *Methods of Information in Medicine*, 22, 93-101.

De Jong, S. (1993), "SIMPLS: an alternative approach to partial least squares regression," *Chemometrics and Intelligent Laboratory Systems*, 18, 251-263.

De'ath, G. (2002), "Multivariate Regression Trees: A new technique for modelling species-environment relationships," *Ecology*, 83, 1105-1117.

De'ath, G., and Fabricius, K. E. (2000), "Classification and Regression Trees: A powerful yet simple technique for ecological data analysis," *Ecology*, 81, 3178-3192.

Dietterich, T. G. (2000a), "Ensemble methods in machine learning," *Lecture Notes In Computer Science*, 1857, 1-15.

Dietterich, T. G. (2000b), "An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, Boosting and Randomization," *Machine Learning*, 40, 139-157.

Dudoit, S., and Fridlyand, J. (2002), "A prediction-based resampling method to estimate the number of clusters in a dataset," *Genome Biology*, 3, 0036.0031-0036.0021.

Dudoit, S., and Fridyland, J. (2003), "Bagging to improve the accuracy of a clustering procedure," *Bioinformatics*, 19, 1090-1099.

E. Dusseldorp, and J. J. Meulman. (2001), "Prediction in medicine by integrating regression trees into regression analysis with optimal scaling," *Methods of Information in Medicine*, 40, 403-409.

Efron, B. (1979), "Computers and the theory of statistics: thinking the unthinkable," *SIAM Review*, 21, 460-480.

Esposito, F., Malerba, D., and Semeraro, G. (1997), "A Comparitive analysis of methods for pruning decison trees," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19, 467-491.

Esposito, F., Malerba, D., Semeraro, G., and Tamma, V. (1999), "The effects of pruning methods on the predictive accuarcy of induced decision trees," *Applied Stochastic Models in Business and Industry*, 15, 277-299.

Everitt, B. (1993), *Cluster Analysis*, London: Arnold Publishers.

Fern, X. Z., and Brodley, C. E. (2004), "Solving Cluster Ensemble Problems by Bipartite Graph Partitioning," *ACM International Conference Proceedings Series*, 69.

Fisher, R. A. (1936), "The use of multiple measurements in taxonomic problems," *Annals of Eugenics*, 7, 179-188.

Fraley, C., and Raftery, A. (2002), "Model Based Clustering, Descriminant Analysis and Density Estimation," *Journal of the American Statistical Association*, 97, 611-631.

Freund, Y., and Shapire, R. (1997), "A decision-theoretic generalization of online learning and an application to boosting," *Journal of Computer Systems and System Sciences*, 55, 119-139.

Friedman, J. (1999), "Stochastic Gradient Boosting," *Technical Report Stanford University*.

Friedman, J. (2001), "Greedy Function Approximation: the gradient boosting machine," *Annals of Statistics*, 29, 1189-1232.

Friedman, J., Hastie, T. J., and Tibshirani, R. J. (2000), "Additive logistic regression: a statistical view of boosting (with discussion)," *Annals of Statistics*, 28, 337-374.

Friedman, J., and Popescu, B. (2003), "Importance Sampled Learning Ensembles," *Technical Report Stanford University*.

Friedman, J. H., and Popescu, B. E. (2005), "Predictive learning via rule ensembles," *Technical Report Stanford University*.

Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (1997), *Bayesian Data Analysis*, Chapman and Hall.

Gifi, A. (1990), *Nonlinear Multivariate Analysis*, John Wiley & Sons.

Giudici, P. (2003), *Applied Data Mining*, John Wiley & Sons.

Gower, J. C. (1971a), "A general coefficient of similarity and some of its properties," *Biometrics*, 27, 857-871.

Gower, J. C. (1975), "Generalized Procrustes Analysis," *Psychometrika*, 40, 33-51.

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

Hancock, T., "R package: 'mct'," (2006), timothy.hancock@jcu.edu.au

Hancock, T., Put, R., Coomans, D., Heyden, Y. V., and Everingham, Y. (2005), "A performance comparison of modern statistical techniques for molecular descriptor selection and retention prediction in chromatographic QSRR studies," *Journal of Chemometrics and Intelligent Laboratory Systems*, 76, 185-196.

Hartigan, J. A. (1975), *Clustering Algorithms*, New York: Wiley.

Hastie, T., Tibshirani, R., and Friedman, J. (2001), *Elements of Statistical Learning*, Springer.

Hastie, T., Tibshirani, R., Narasimhan, B., and Chu, G., "R contributed package: 'impute'," (2005), www.r-project.org

Hastie, T. J., Buja, A., and Tibshirani, R. J. (1995), "Penalized discriminant analysis," *Annals of Statistics*, 23, 73-102.

Hoerl, A. E., and Kennard, R. W. (1970), "Ridge Regression: Biased estimation for nonorthogonal problems," *Technometrics*, 13, 55-67.

Hong, N. T. (1997), "Trace Element Analysis With Application To Environmental Pollutants Studies In Vietnam," *Doctor of Philosophy Dissertation, Chalmers University of Technology and Goteborg University. S-412 96 Goteborg Sweden*.

Hornik, K., "R contributed package: 'clue'," (2006), www.r-project.org

Hubert, L., and Arabie, P. (1985), "Comparing Partitions," *Journal of Classification*, 2, 193-218.

Karypis, G., and Kumar, V. (1998), "A fast and high quality multilevel scheme for partitioning irregular graphs," *SIAM Journal of Scientific Computing*, 20, 359-392.

Kass, G. V. (1980), "An Exploratory Technique for Investigating Large Quantities of Categorical Data," *Applied Statistics*, 29, 119-127.

Kaufman, L., and Rousseeuw, P. J. (1987), *Clustering by means of medoids, in Statistical Data Analysis based on the L1 Norm*, ed. Y. Dodge, Amserdam: Elsevier.

Kaufman, L., and Rousseeuw, P. J. (1990), *Finding groups in data*, John Wiley & Sons.

Kavsek, B., Lavrac, N., and Ferligoj, A. (2001), "Consensus Decision Trees: Using hierarchical clustering for data relabelling and reduction," *Proceedings of the 12th European Conference on Machine Learning*, 2167, 251-262.

Larsen, D. R., and Speckman, P. L. (2004), "Multivariate regression trees for analysis of abundance data," *Biometrics*, 60, 534-549.

Lazzeroni, L., and Owen, A. (2002), "Plaid Models for Gene Expression Data," *Statistical Sinica*, 12, 61-86.

Lebart, L., Morneau, A., and Warwick, K. M. (1984), *Multivariate Descriptive Statistical Analysis: Correspondence analysis and related techniques for large matrices*, John Wiley & Sons.

Leisch, F., and Dimitriadou, E., "R contributed package: 'mlbench'," (2005), www.r-project.org

Lent, B., Swami, A., and Widom, J. (1997), "Clustering Association Rules," *Proceedings of International Conference on Data Enginneering*, 220-231.

Maechler, M., Rousseeuw, P., Struyf, A., Hubert, M., and Hornik, K., "R contributed package: 'cluster' (v. 1.10.4)," (2006), www.r-project.org

Mangasrian, O. L., and Wolberg, W. H. (1990), "Cancer diagnosis via linear programming," *SIAM News*, 23.

Mclachlan, G. J., Bean, R. W., and Peel, D. (2002), "A mixture model-based approach to the clustering of microarray expression data," *Bioinformatics*, 18, 413-422.

Mcleish, M., and Cecile, M. (1989), "Horse Colic Dataset," *http://lib.stat.cmu.edu*.

Milligan, G. W., and Cooper, M. C. (1985), "An examination of procedures for determining the number of clusters in a dataset," *Psycometrika*, 50, 159-179.

Mitchell, M. (1998), *An Introduction To Genetic Algorithms* (1998 ed.), The MIT Press.

Monti, S., Tamayo, P., Mesirov, J., and Golub, T. (2003), "Consensus Clustering: A Resampling-Based Method for Class Discovery and Visualisation of Gene Expression Microarray Data," *Machine Learning*, 52, 91-118.

Pantazi, S., Kagolovsky, Y., and Moehr, J. R. (2002), "Cluster Analysis of Wisconsin Breast Cancer Dataset Using Self-Organising Maps," *MIE2002, Budapest, Hungry*.

Questier, F., Put, R., Coomans, D., Walczak, B., and Vander Heyden, Y. (2004), "The use of CART and multivariate regression trees for supervised and unsupervised feature selection," *Chemometrics and Intelligent Laboratory Systems*, 76, 45-54.

Quinlan, J. R. (1986), "Induction of decision trees," *Machine Learning*, 1, 81-106.

Quinlan, J. R. (1987), "Simplifying decision trees," *International Journal of Man-Machine Studies*, 27, 221-234.

Quinlan, R. (1993), *C4.5: Programs for machine learning*, San Mateo: Morgan Kaufmann.

R Development Team, "R: A language and environment for statistical computing," (2005), http://www.r-project.org/

Raftery, A. E., Madigan, D., and Hoeting, J. (1997), "Bayesian Model Averaging for Linear Regression Models," *Journal of the American Statistical Association*, 92, 179-191.

Rencher, A. C. (2002), *Method of Multivariate Analysis*, John Wiley & Sons.

Sain, S. R., and Carmack, P. S. (2002), "Boosting multi-objective regression trees," *Computing Science and Statistics*, 34, 232-241.

Schapire, R. E. (1990), "The strength of weak learnability," *Machine Learning*, 5, 197-227.

Schapire, R. E. (2002), "The boosting approach to machine learning: An overview," *In MSRI Workshop on Nonlinear Estimation and Classification, Berkeley, CA, Mar. 2001. http://stat.bell-labs.com/who/cocteau/ nec/ and http://www.research.att.com/~schapire/boost.html*.

Schapire, R. E., Freund, Y., Bartlett, P., and Lee, W. S. (1998), "Boosting The Margin: A new explanation for the effectiveness of voting methods," *Annals of Statistics*, 26, 1651-1686.

Schork, M. A., and Remington, R. D. (2000), *Statistics with Applications To The Biological and Health Sciences* (Third ed.), Prentice Hall.

Seber, G. A. F. (1984), *Multivariate Observations*, John Wiley & Sons.

Segal, M. R. (1992), "Tree-structured methods for longitudinal data," *Journal of the American Statistical Association*, 87, 407-418.

Seong Keon Lee, Hyun-Cheol Kang, Sang-Tae Han, and Kwang-Hwan Kim. (2005), "Using Generalized Estimating Equations to Learn Decision Trees with Multivariate Responses," *Data mining and knowledge discovery*, 11, 273-293.

Shi, T., and Horvath, S. (2003), "Using random forests similarities in unsupervised learning: Applications to microarray data," *Atlantic Symposium on Computation Biology and Genome Informaticcs (CBGI'03)*.

Shi, T., and Horvath, S. (2006), "Unsupervised Learning with Random Forest Predictors," *Journal of Computation and Graphical Statistics*, 15, 118-138.

Smyth, C., Coomans, D., and Everingham, Y. (2006), "Clustering Noisy Data In A Reduced Dimensional Space via Multivariate Regression Trees," *Pattern Recognition*, 39, 424-431.

Smyth, C., Coomans, D., Everingham, Y., and Hancock, T. (2005), "Auto-associative Multivarite Regression Trees for Cluster Analysis," *Chemometrics and Intelligent Laboratory Systems*, 80, 120-129.

Srikant, R., and Agrawal, R. (1997), "Mining Generalized Association Rules," *Future Generation Computer Systems*, 13, 161-180.

Stephen Swift, Allan Tucker, Veronica Viniotti, Nigel Martin, Christine Orengo, Xiaohui Liu, and Paul Kellam. (2004), "Consensus clustering and functional interpretation of gene-expression data," *Genome Biology*, 5(11):R94.

Strehl, A. (2002), "Relationship-based Clustering and Cluster Ensembles for High-dimensional Data Mining," *Doctor of Philosophy Dissertation; The University of Texas*.

Strehl, A., and Ghosh, J. (2002), "Cluster Ensembles - A Knowledge Reuse Framework for Combining Multiple Partitions," *Journal of Machine Learning Research*, 3, 583-617.

Therneau, T. M., Atkinson, B., and Ripley, B., "R contributed pacakge: 'rpart'," (2005), www.r-project.org

Therneau, T. M., Atkinson, B., Ripley, B., and De'ath, G., "R contributed package: 'mvpart'," (2004), www.r-project.org

Tibshirani, R. J., Walther, G., Botstein, D., and Brown, P. (2005), "Cluster Validation by Prediction Strength," *Journal of Computation and Graphical Statistics*, 14, 511-238(518).

Tibshirani, R. J., Walther, G., and Hastie, T. J. (2001), "Estimating the number of clusters in a dataset via the gap statistic," *Journal of the Royal Statistical Society B*, 63, 411-424.

Topchy, A., Jain, A. K., and Punch, W. (2004), "A Mixture Model for Cluster Ensembles," *Proceedings of the Fourth SIAM International Conference 2004 (SDM 2004)*.

Topchy, T., Minaei-Bidgoli, B., Jain, A. K., and Punch, W. F. (2004), "Adaptive Cluster Ensembles," *ICPR 2004*, 272-275.

Torgerson, W. S. (1958), *Theory and Methods of Scaling*, New York: Wiley.

Weingessel, A., Dimitriadou, E., and Hornik, K. (2001), "An Ensemble Method for Clustering," *In Proceedings of the International Conference on Artificial Neural Networks, Vienna (2001)*, 217-224.

Wolberg, W. H., and Mangasrian, O. L. (1990), "Multisurface method of pattern separation for medical diagnosis applied to breast cytology," *Applied Mathematics*, 87, 9193-9196.

Yan Yu, and Diane Lambert. (1999), "Fitting Trees to Functional Data With an Application to Time-of-Day Patterns," *Journal of Computation and Graphical Statistics*, 8, 749-762.

Yeung, K. Y., Haynor, D. R., and Ruzzo, W. L. (2001), "Validating clustering for gene expression data," *Bioinformatics*, 17, 309-318.

Young, F. W. (1981), "Quantitative Analysis of Qualitative Data," *Psychometrika*, 46, 357-388.

Zhang, H. (1998), "Classification Trees for Multiple Binary Responses," *Journal of the American Statistical Association*, 98, 180-193.