Proceedings of the 2007 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB 2007)

Clustering Microarrays with Predictive Weighted Ensembles

Christine Smyth and Danny Coomans
Statistics and Intelligent Data Analysis Group
School of Mathematics, Physics and Information Technology
James Cook University
Townsville, 4811, Australia

***Abstract*-Cluster ensembles seek a consensus across many individual partitions and the resulting solution is usually stable. Cluster ensembles are well suited to the analysis of DNA microarrays, where the tremendous size of the dataset can thwart the discovery of stable groups. Post processing cluster ensembles, where each individual partition is weighted according to its relative accuracy improves the performance of the ensemble whilst maintaining its stability. However, weighted cluster ensembles remain relatively unexplored, primarily because there are no common means of assessing the accuracy of individual clustering solutions. This paper describes a technique of creating weighted cluster ensembles suitable for use with microarray datasets. A regression technique is used to obtain individual cluster solutions. Each solution is then weighted according to its predictive accuracy. The consensus partition is obtained using a novel modification to the traditional k-means algorithm which further enforces the predictability of the solution. An estimate of the natural number of clusters can also be obtained using the modified k-means algorithm. Furthermore, a valuable byproduct of this weighted ensemble approach is a variable importance list. The methodology is applied on two well-known microarray datasets with promising results.**

I. Introduction

Cluster analysis plays a vital role in the understanding of large DNA microarrays. However, the large number of variables in these datasets can cloud the underlying groups, and traditional clustering algorithms may produce inaccurate or unstable results. This motivates the application of cluster ensembles to DNA microarrays. Cluster ensembles seek a consensus across many individual clustering solutions, often grown on smaller subsets of the data, with the aim of finding a stable partition.

Cluster ensembles combine individual solutions in various ways. A common approach involves the creation of a co-occurrence matrix for each clustering solution. Basically, the $(i, j)^{th}$ element of the co-occurrence matrix equals one if observational units i and j are clustered together by the algorithm and zero otherwise. The co-occurrence matrices of each model within the ensemble are aggregated to give an overall co-occurrence matrix, where the $(i, j)^{th}$ element represents the percentage of times observational units i and j are clustered together. The overall matrix is a similarity matrix and can be split using a variety of clustering techniques, such as hierarchical clustering or partitioning around medoids [1].

However, within a cluster ensemble there will be both "good" and "bad" partitions [2]. Assigning low weights to inaccurate co-occurrence matrices, and then taking a weighted aggregation of the individual co-occurrence matrices should improve the performance of the cluster ensemble. However, weighting (post processing) individual clustering solutions within an ensemble remains relatively unexplored. Unlike regression and classification ensembles, where the accuracy of individual models can easily be gauged using a loss criterion between the predicted values and the observed response, there are no criteria suitable for assessing individual clustering solutions within an ensemble.

Previously, we suggested a technique of weighting cluster ensembles for small datasets [3]. The accuracy of each clustering solution was assessed on the basis of its predictive error. The weighted cluster ensembles outperformed simple average ensembles and individual clustering models. Here, we apply the technique with some modifications for large datasets to DNA microarrays.

We propose that by using a regression technique, multivariate regression trees, as a clustering algorithm, each solution can be assessed according to its predictive accuracy. Previous literature has shown that multivariate regression trees double effectively as a clustering technique [4],[5]. If clustering in a low dimensional setting, the explanatory variables are replicated as the response variables (auto-associative multivariate regression tree), and the clusters are found in the entire variable space. If clustering in a high dimensional setting, the dimension is first reduced using principal components analysis or factor analysis, and the resulting scores are used as the response variables [6]. The response set can be made as small as desired by taking the first q principal components or factors. Searching in the reduced dimension space for clusters is particularly appealing when analyzing DNA microarrays where some variables serve only to distort the underlying grouping structure.

By sampling explanatory variables, many trees can be grown. Trees with high predictive accuracy are then given large weights. The weighting procedure can easily be performed using any well-known regression post processing technique. Here we use the forward stagewise approximation [7] to the lasso [8].

By taking the co-occurrence matrix given by each tree and multiplying it by the tree's weight, and then summing the weighted co-occurrence matrices together, an overall "weighted co-occurrence matrix" is obtained. This co-

occurrence matrix can be considered the output of a weighted cluster ensemble approach. The approach assumes that trees with high predictive accuracy produce "good" clusters. Using predictive accuracy to assess cluster quality has previously been suggested [9],[10],[11].

To partition the weighted co-occurrence matrix we introduce similarity-based k-means (SBK) [3]. SBK enforces the predictability of the solution by explicitly predicting the group structure found within the entire similarity matrix (including the covariance submatrices) shown to be important in [12]. An approximation to the natural number of clusters in the dataset can also be obtained with SBK using a technique modeled on [10].

Furthermore, the underpinning weighted ensemble produces a list of variables (genes) that are important in differentiating the clusters. A variable importance list gives experimentalists an idea of genes that may warrant further investigation as potential biomarkers particularly if the genes are differentiating between two groups (say cancer versus non-cancer).

We illustrate the weighted cluster ensemble approach on two well-known DNA microarray datasets. The clustering results are consistent with others in the literature. Some genes in the derived variable importance lists are known to be important in classifying the groups within the datasets. The estimates of the natural number of clusters tend to agree with the known number of classes in the data.

II. THEORY

A. Multivariate Regression Trees

Regression trees [13],[14] begin with all the data in one node. At each stage, the regression tree partitions a non-split node in two. Regression trees partition a node, t, into two subsets, t_L and t_R, on the basis of the value of an explanatory variable. At each node all possible splits of each explanatory variable are considered. The optimal split is saved for each node. The node with the split that maximizes the decrease in $R(T)$ is partitioned at each stage. $R(T)$ is given by:

$$R(T) = \frac{1}{n}\sum_{t\in\tilde{T}}\sum_{\underset{\sim}{x}_i\in t}\left(\underset{\sim}{y}_i - \underset{\sim}{\bar{y}}(t)\right)^T\left(\underset{\sim}{y}_i - \underset{\sim}{\bar{y}}(t)\right) \quad (1)$$

where $\underset{\sim}{x}_i$ is the vector of measurements of P explanatory variables for the i^{th} observational unit; $\underset{\sim}{y}_i$ is the vector of measurements of the response variables for the i^{th} observational unit; $\tilde{T}$ is the set of all terminal nodes and; $\underset{\sim}{\bar{y}}(t)$ is the mean response vector of terminal node t.

After growing the tree, the non-split nodes are deemed "terminal". The predicted value for a terminal node, t_{term}, is:

$$\underset{\sim}{\hat{y}}(t_{term}) = \frac{1}{n_{t_{term}}}\sum_{\underset{\sim}{x}_i\in t_{term}}\underset{\sim}{y}_i \quad (2)$$

where the sum is over all $\underset{\sim}{y}_i$ such that $\underset{\sim}{x}_i \in t_{term}$ and $n_{t_{term}}$ is the total number of cases in the terminal node.

The observational units in each of the terminal nodes are the clusters of the dataset: the terminal nodes are as homogeneous as possible reflecting an intuitive definition of a cluster. The clusters are found in the response space and the explanatory variables that form the tree are deemed to be important in determining the clusters. To allow multivariate regression trees to be applied in the traditional clustering framework where there are no response variables, auto-associative multivariate regression trees (AAMRTs) were suggested [4],[5]. AAMRTs replicate the explanatory variables as response variables and grow the tree using identical response and explanatory datasets.

If the number of the variables is too large, AAMRTs may be confused by the redundant or 'noise' variables and may produce inaccurate results. To overcome this flaw, the dimension of the response space can be reduced using either principal components or factor analysis. Principal components analysis attempts to model the total variance of the original dataset, via new uncorrelated variables called principal components. Factor analysis attempts to explain the variables by assuming that they can be generated as a linear combination of q unobservable common factors (usually $q << P$) plus a unique factor [15]. We use either the principal component scores from the first q principal components or the factor scores from the q factors as the response variables of the tree [6]. The clustering is obtained in the reduced dimension space as q is less than P. Trees grown using principal component scores are referred to as MRTPCs, and similarly, trees grown using factor scores are referred to as MRTFSs.

B. Algorithm for creating ensembles of AAMRTs, MRTPCs, or MRTFSs

Algorithm 1 shows the process used to grow an ensemble of regression trees such that they can be used to create a cluster ensemble.

Algorithm 1: Growing an ensemble of trees

1) Choose the number of individual trees in the ensemble, M.

2) If the trees in the ensemble are AAMRTs, replicate the original dataset as the response dataset, Y. If the trees in the ensemble are MRTPCs calculate the first q principal component scores as the response dataset, Y. Or, if the trees in the ensemble are MRTFSs, calculate the first q factor scores as the response dataset, Y. The choice of q is left to the investigator.

3) Create M explanatory datasets by randomly sampling variables with percentage p_v from the original dataset. Here we use $p_V = 0.05$. We stress that although the variables may be sampled to create different explanatory datasets, the response for each tree remains constant.

4) Grow a tree using each explanatory dataset to k terminal nodes (clusters). Create a co-occurrence matrix for each tree, $C(m)$, where the $(i,j)^{th}$ element of the matrix is one, if observational units i and j are in the same cluster (terminal node). Create a variable importance list for each tree using Algorithm 5.

C. Lasso Heuristic

A regression ensemble can be represented by:

$$F(\underset{\sim}{x}) = \sum_{m=1}^{M} \omega_m f_m(\underset{\sim}{x}) \quad (3)$$

where $f_m(\underset{\sim}{x})$ is the prediction of an observational unit $\underset{\sim}{x}$ by the m^{th} model - the $f_m(\underset{\sim}{x})$ are usually of the same family of models but this is not mandatory; ω_m is the weight assigned to $f_m(\underset{\sim}{x})$; and M is the number of models.

The individual regression solutions are combined to form an ensemble by taking a weighted sum of the individual solutions. Usually, the weights are an average of the number of models, $1/M$, see for example [7]. Post processing is a procedure which suggests choices of ω_m that reflect the relevance of each $f_m(\underset{\sim}{x})$ [16]. Post processing usually achieves greater accuracy by enforcing parsimony. The lasso [8] post processing procedure finds the weights that minimize

$$\hat{\underset{\sim}{\omega}} = \arg\min_{\underset{\sim}{\omega}} \sum_{i=1}^{n} \left(\underset{\sim}{y}_i - \sum_{m=1}^{M} \omega_m f_m(\underset{\sim}{x}_i) \right)^T \left(\underset{\sim}{y}_i - \sum_{m=1}^{M} \omega_m f_m(\underset{\sim}{x}_i) \right) + \lambda \sum_{m=1}^{M} |\omega_m|. \quad (4)$$

Here, the solution to the lasso is approximated with a forward stagewise algorithm [7] which is henceforth referred to as the "lasso heuristic". The algorithm is as follows:

Algorithm 2: The lasso heuristic

1) Set all weights to zero. Choose ε as a small number greater than zero, and choose the number of iterations, *its*, to be quite large.

2) For $l = 1 : its$

$$(\beta^*, h^*) = \arg\min_{\beta, h} \sum_{i=1}^{n} \left(\underset{\sim}{y}_i - \sum_{m=1}^{M} \omega_m f_m(\underset{\sim}{x}_i) - \beta \times f_h(\underset{\sim}{x}_i) \right)^T \left(\underset{\sim}{y}_i - \sum_{m=1}^{M} \omega_m f_m(\underset{\sim}{x}_i) - \beta \times f_h(\underset{\sim}{x}_i) \right). \quad (5)$$

$$\hat{\omega}_{h^*} = \hat{\omega}_{h^*} + \varepsilon \times sign(\beta^*). \quad (6)$$

3) Finally,

$$F(\underset{\sim}{x}) = \sum_{m=1}^{M} \hat{\omega}_m f_m(\underset{\sim}{x}). \quad (7)$$

In the first step all weights are zero, and this is analogous to $\lambda = \infty$ in (4). The parameter *its* is inversely related to λ in (4). After the set number of iterations, many weights will still remain zero.

D. Algorithm for producing a weighted co-occurrence matrix

Algorithm 3 shows the process used to create a weighted co-occurrence matrix.

Algorithm 3: Producing a weighted co-occurrence matrix

1) Create an ensemble of trees using Algorithm 1.

2) Post process the ensemble of regression trees to find the weights using Algorithm 2. Here, $f_m(\underset{\sim}{x}_i)$ is the prediction of observational unit i using the m^{th} regression tree. The response vector, $\underset{\sim}{y}_i$ is given by: $\underset{\sim}{x}_i$ if using AAMRTs or; the associated vector of q principal component scores if using MRTPCs or; the associated vector of q factor scores if using MRTFSs.

3) Create an overall co-occurrence matrix, C by taking a weighted sum of the individual co-occurrence matrices:

$$C = \sum_{m=1}^{M} \hat{\omega}_m C(m). \quad (8)$$

Taking a weighted sum of dissimilarity matrices created from different sources (where the weights were chosen in a "subjective way") was suggested previously by [17].

E. Similarity-based k-means

Similarity-based k-means is a divisive clustering algorithm that takes a co-occurrence matrix, C, (similarity matrix) as input. Formally, SBK seeks clusters to minimize either of the objective functions:

$$\min \sum_{r=1}^{k} \sum_{i,j \in S_r} \left(C_{i,j} - \bar{C}_r \right)^2 + \sum_{r=1}^{k} \sum_{\substack{r'=1 \\ r' \neq r}}^{k} \sum_{i \in S_r} \sum_{j \in S_{r'}} \left(C_{i,j} - \overline{COV}_{(S_r, S_{r'})} \right)^2 \quad (9)$$

or

$$\min \sum_{r=1}^{k} \sum_{i,j \in S_r} \left| C_{i,j} - \bar{C}_r \right| + \sum_{r=1}^{k} \sum_{\substack{r'=1 \\ r' \neq r}}^{k} \sum_{i \in S_r} \sum_{j \in S_{r'}} \left| C_{i,j} - \overline{COV}_{(S_r, S_{r'})} \right| \quad (10)$$

where k is the number of clusters; i, j index observational units i and j; S_r is the set of observational units in the r^{th} cluster; $C_{i,j}$ is the $(i,j)^{th}$ element of the co-occurrence matrix; $\bar{C}_r$ is the mean similarity of the r^{th} cluster; and $\overline{COV}_{(S_r, S_{r'})}$ is the mean similarity of the (covariance) matrix where the rows are given by the observational units in cluster r and the columns are dictated by the observational units in the r'^{th} cluster. The covariance submatrices should be considered the number of times that observational units in one cluster are grouped with observational units in another cluster during the ensemble creation.

Because of the mean squared and absolute error terms in the objective functions (9) and (10), SBK can be viewed almost entirely in the prediction sense. The algorithm seeks to predict the entire co-occurrence matrix using the cluster and covariance means. In doing so, observational units with high similarity are grouped together. A validity criterion [12] is imposed to ensure that the clustering ideology prevails over the prediction ideology. The validity criterion dictates that clusters must have higher mean similarities than their covariance matrices. The SBK algorithm is given by Algorithm 4.

Algorithm 4: SBK

1) Choose the number of clusters and an initial partition of the data. Here, we use initial partitions given by both hierarchical clustering of the co-occurrence matrix and entirely random partitions. Choose the objective function; either the mean squared error (9) or absolute error (10).

2) Visit each observational unit and assign it to the cluster which will result in the largest decrease of the objective

function. Before moving the observational unit ensure that the validity criterion is upheld.
3) Update the mean similarity of: the cluster the observational unit has left; the cluster the observational unit has joined; and all appropriate covariance means.
4) Repeat steps two and three until no more reassignments of the observational units take place.

F. Cluster number estimation

An approximation to the natural number of clusters in the dataset can also be obtained with SBK by considering the average predictive capability of the algorithm, for any number of clusters, k. The estimate closely resembles the figure of merit (FOM) method proposed by [10]. FOMs are a method of authenticating clusters by assessing the "predictive power" of a clustering technique. FOMs require no a priori knowledge of group membership. FOMs have been shown to provide an accurate estimate of the natural number of clusters [5],[10]. FOMs assess the "predictive power" of a clustering algorithm by leaving out a variable p, clustering the data (into k clusters), then calculating the root mean square error (RMSE) of p relative to the cluster means:

$$RMSE(p,k) = \sqrt{\frac{1}{n}\sum_{r=1}^{k}\sum_{x_i \in S_r}\left(x_{ip} - \bar{x}_r(p)\right)^2} \qquad (11)$$

where x_{ip} is the measurement of the p^{th} variable on the i^{th} observational unit; n is the number of observational units; S_r is the set of observational units in the r^{th} cluster; $\bar{x}_r(p)$ is the mean of variable p for the observational units in the r^{th} cluster.

Each variable is omitted and its RMSE calculated. These RMSEs are summed over all variables to give an aggregate FOM (AFOM):

$$AFOM(k) = \sum_{p=1}^{P} RMSE(p,k). \qquad (12)$$

The AFOM is calculated for each k, and adjusted for cluster size to give $AFOM_{adj}(k)$.

It is simple to expand the above AFOM theory to the results obtained by SBK. Here, no variables are removed from the dataset; the random nature of SBK introduces enough variability. Simply, if the dataset is clustered into k clusters and this process is repeated P times, then the $AFOM(k)$ is defined as

$$AFOM(k) = \sum_{p=1}^{P}\sqrt{\frac{1}{n^2}\sum_{r=1}^{k}\sum_{i,j \in S_r(p)}\left(C_{i,j} - \bar{C}_r(p)\right)^2} \qquad (13)$$

where $S_r(p)$ is the set of observational units in cluster r on the p^{th} run; $\bar{C}_r(p)$ is the mean similarity of the observational units in cluster r on the p^{th} run; $C_{i,j}$ is the $(i,j)^{th}$ element of the co-occurrence matrix; and n^2 is the dimension of the similarity matrix. Here, the adjusted figure of merit is given by:

$$AFOM_{adj}(k) = \frac{AFOM(k)}{P\sqrt{\frac{n^2-k}{n^2}}}. \qquad (14)$$

The $AFOM_{adj}$ is obtained for varying levels of k and the "elbow" of the graph is selected as the optimal number of clusters.

G. Variable Importance

Multivariate regression trees allow for an easy calculation of a variable importance list. Although many definitions of variable importance exist, such as those that consider surrogate splits [13], we apply a very simple (but naive) definition of variable importance. Our definition of variable importance, if applied to only one tree grown on the entire dataset would over-inflate the importance of some variables and underestimate the importance of others. However, our reasoning is that the random sampling of variables to build each tree will give some stability to our variable importance list that would otherwise not exist. We calculate a variable importance list for each tree in the ensemble using Algorithm 5. The variable importance list for the entire ensemble is then the weighted sum of the variable importance lists for each tree, using the weights given by the lasso heuristic of Algorithm 2.

Algorithm 5: Variable importance list for a single tree
1) For each variable, p sum the $\Delta R(t)$ for all splits made by that variable within the tree to obtain the variable importance of p, VI_p. Mathematically, VI_p is given by:

$$VI_p = \sum_{\substack{t \in T \text{ where} \\ \text{the node is} \\ \text{split by } p}} \Delta R(t) \qquad (15)$$

where

$$R(t) = \sum_{\underset{\sim}{x}_i \in t}\left(\underset{\sim}{y}_i - \underset{\sim}{\bar{y}}(t)\right)^T\left(\underset{\sim}{y}_i - \underset{\sim}{\bar{y}}(t)\right) \qquad (16)$$

and

$$\Delta R(t) = R(t) - (R(t_L) + R(t_R)) \qquad (17)$$

and t designates the parent node; and t_L and t_R designate the left and right nodes respectively.
If a variable is not included in the predictor set of a particular tree, its corresponding variable importance for the tree is zero.
2) Standardize the variable importance for the tree such that the individual importances sum to one.

H. Cluster Evaluation

Assessing the validity and accuracy of clustering algorithms is not a straightforward task. Various algorithms have been suggested in the recent literature, see for example [10] and [18]. However, in this paper we use datasets with known classifications and we assume these to be the gold standard. As such, we report only the number of "misclassifications" (similar to [9] and [19]), but recognize that in real world settings this is not possible.

III. DATA

Two well known microarray datasets were analyzed. The reader is directed to the references for detailed information regarding these datasets. The first dataset, 'Alon' [20], contains 62 samples measured on 2000 genes. There are 22 samples of normal colon tissue, and 40 samples of tumor tissue. The 100 variables with the largest variance were used in this analysis. The dataset, available from the R package 'dprep' [21], was preprocessed by taking the logarithm (base 10), and standardizing the tissues and genes to have zero mean and unit standard deviation.

The second dataset, 'Golub' [22], contains 72 samples measured on 6817 genes. The number of genes was decreased to 3571 using the steps of [23]. There are 47 samples of Acute Lymphoblastic Leukemia (ALL) and 25 samples of Acute Myeloid Leukemia (AML). The ALL class can be further divided into two subgroups consisting of 38 B-cell ALL and 9 T-cell ALL. The 100 variables with the largest variance were used in this analysis in the same manner as [23]. The dataset, (which has already been log transformed and row standardized) available from the R package 'dprep', was preprocessed by standardizing the genes to have unit standard deviation.

IV. PROCEDURE

The individual tree models in the ensemble require the specification of the number of terminal nodes and the minimum terminal node size. To assess the sensitivity of the results to varying values of these parameters, we ran Algorithm 1 with either (1,5,10) minimum terminal node size and either (2,4,6) terminal nodes. There were 3*3=9 choices of parameters and an ensemble of trees was grown for each choice. We also grew an ensemble with random inputs to the parameters. Each tree within the ensemble was randomly assigned a minimum terminal node size and number of terminal nodes from the above sets. This resulted in a total of 10 ensembles being grown for each of AAMRTs, MRTFSs, and MRTPCs. There were therefore 30 ensembles created for each dataset. All ensembles were grown to 500 trees. The parameter q was taken to be 10.

The M co-occurrence matrices for a set of parameters and response type were weighted using Algorithms 2 and 3. The weighted co-occurrence matrices were then split using SBK. When splitting co-occurrence matrices the minimization criteria (9) and (10) of SBK were used and both results are shown. The results reported were the most frequently occurring using 15 different starting points. The datasets were split to a maximum of 10 clusters so that the AFOM graphs could be obtained. However, the reported results are those when the co-occurrence matrix was split to the known number of groups in the data. Variable importance lists were also obtained. All analysis was conducted using [24].

V. RESULTS

A. Alon Dataset

The results of applying SBK to the weighted co-occurrence matrices created by each of the ensemble types are reported in Table I. The first row shows the number of terminal nodes of the trees in the ensemble. The second row shows the minimum terminal node size of the trees in the ensemble. The 'R' in both the first and second rows corresponds to the ensembles of trees grown on random parameter (minimum terminal node size and number of terminal nodes) values. The types of trees in the ensemble are shown in the final three rows. The reader will see that there are ten ensembles grown for each response type. The number of misclassifications using SBK with (9) is shown as the top number of the cell, and the number of misclassifications using (10) is shown as the bottom number of the cell.

The results of applying SBK to the co-occurrence matrices created by the ensembles of AAMRTs and MRTPCs are fairly consistent across minimum terminal node size and number of terminal nodes. The misclassification rates of SBK applied to the co-occurrence matrices created by the AAMRT and MRTPC ensembles grown with random parameters are a fair compromise of the misclassifications using set parameters. The misclassification rates of applying SBK to the co-occurrence matrices created by ensembles of MRTFSs are less stable than the other two ensemble types.

The AFOM graphs tend to indicate that the weighted co-occurence matrices should be split to three clusters. A sample AFOM graph (with error bars) is shown in Fig. 1. It was obtained by applying SBK with (9) to a weighted co-occurrence matrix constructed by MRTPCs with random parameters. Growing to three clusters improves the results considerably as shown in Table II. There is a high degree of similarity between the misclassification rates of applying SBK to the co-occurrence matrices of the ensembles of AAMRTs and MRTPCs. Growing these ensembles with random parameters gives a compromise of the misclassifications using the set parameters. On the other hand, using random parameters with ensembles of MRTFSs does not give solutions that are representative of ensembles with set parameters.

The top five important variables using each response type are presented in Table III. The variables are presented in decreasing order of importance. There is a degree of overlap between the response types, particularly using ensembles of AAMRTs and MRTPCs.

TABLE I

NUMBER OF MISCLASSIFICATIONS FOR THE ALON DATASET – TWO CLUSTERS

Number of terminal nodes	2			4			6			R
Minimum terminal node size	1	5	10	1	5	10	1	5	10	R
AAMRT	9	9	13	15	15	7	14	15	7	13
	14	14	14	15	15	10	15	15	10	13
MRTPC	14	14	9	15	16	7	16	6	7	13
	13	13	13	15	15	10	15	13	10	12
MRTFS	22	22	22	7	22	10	9	6	10	12
	22	22	22	8	13	9	13	6	9	10

TABLE II

NUMBER OF MISCLASSIFICATIONS FOR THE ALON DATASET – THREE CLUSTERS

Number of terminal nodes	2			4			6			R
Minimum terminal node size	1	5	10	1	5	10	1	5	10	R
AAMRT	10	10	10	9	9	6	7	7	6	8
	10	10	10	10	10	10	8	11	10	9
MRTPC	10	10	10	9	9	7	7	6	7	8
	10	10	10	9	9	10	8	8	10	8
MRTFS	8	7	10	7	6	7	6	6	7	13
	8	7	8	9	9	10	6	6	10	12

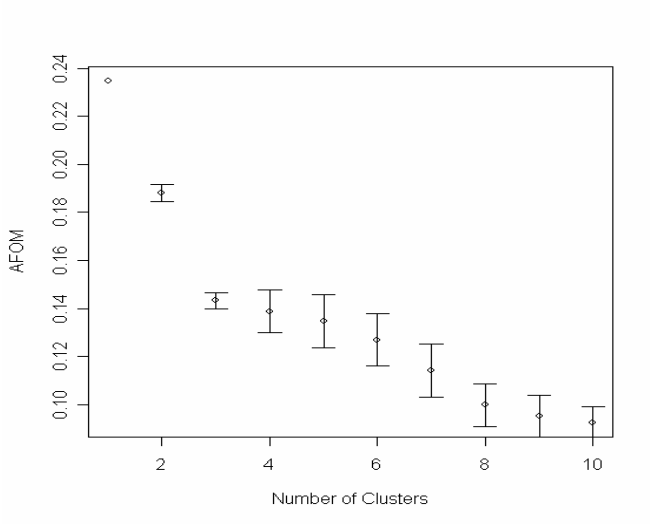

Fig. 1. AFOM graph for the Alon dataset.

TABLE III

IMPORTANT VARIABLES FOR THE ALON DATASET

Ensemble	Gene description
AAMRT	Human 11 beta-hydroxysteroid dehydrogenase type II mRNA, complete cds
	ACTIN, AORTIC SMOOTH MUSCLE (HUMAN)
	H. sapiens mRNA for hevin like protein
	P24050 40S RIBOSOMAL PROTEIN
	Human mRNA for fibronectin (FN precursor)
MRTPC	Human 11 beta-hydroxysteroid dehydrogenase type II mRNA, complete cds
	ACTIN, AORTIC SMOOTH MUSCLE (HUMAN)
	H. sapiens mRNA for hevin like protein
	PUTATIVE 126.9 KD TRANSCRIPTIONAL REGULATORY PROTEIN IN YSW1-RIB7 INTERGENIC REGION (Saccharomyces cerevisiae)
	TRANSLATIONAL INITIATION FACTOR 2 BETA SUBUNIT (HUMAN)
MRTFS	P24050 40S RIBOSOMAL PROTEIN
	Human CO-029
	Human 11 beta-hydroxysteroid dehydrogenase type II mRNA, complete cds
	H. sapiens mRNA for novel DNA binding protein
	SELENIUM-BINDING PROTEIN (Mus musculus)

B. Golub Dataset

The results of splitting the weighted co-occurrence matrices created by each of the tree types are shown in Table IV. The misclassification rates using SBK on co-occurrence matrices created by ensembles of AAMRTs and MRTPCs are similar. Using these two types of trees with random parameters also gives misclassification rates that are representative of the set parameters. Again, SBK applied to the co-occurrence matrices created by MRTFSs does not produce as stable results as with the other two types of trees.

The AFOM graphs indicate splitting to three clusters will produce the optimal results. A sample AFOM graph is shown in Fig. 2. The graph was obtained by applying SBK with (9) to the weighted co-occurrence matrix constructed by AAMRTs with a minimum terminal node size of five and two terminal nodes.

Splitting the weighted co-occurrence matrices uncovers the three known subgroups in the data. The misclassification rates are shown in Table V. The table may indicate that if the minimum terminal node size of the trees is too large, the misclassification rates of SBK suffer. Again, splitting the co-occurrence matrices created by ensembles of AAMRTs and MRTPCs produces similar, stable results. However, splitting the co-occurrence matrices of ensembles of MRTFSs using SBK produces unstable results across set tree parameters. Also, the results are not indicative of the set parameters when the trees use random parameters.

The top five variables using each tree type are shown in Table VI. Again, there is a large degree of overlap amongst the ensembles of AAMRTs and MRTPCs.

TABLE IV

NUMBER OF MISCLASSIFICATIONS FOR THE GOLUB DATASET – TWO CLUSTERS

Number of terminal nodes	2			4			6			R
Minimum terminal node size	1	5	10	1	5	10	1	5	10	R
AAMRT	9	9	9	10	10	2	10	10	4	9
	9	9	9	10	10	9	10	9	9	9
MRTPC	9	9	9	10	10	10	10	10	4	9
	9	9	9	10	9	10	10	10	10	9
MRTFS	13	13	17	7	7	11	10	12	12	10
	13	13	17	19	19	9	10	11	5	10

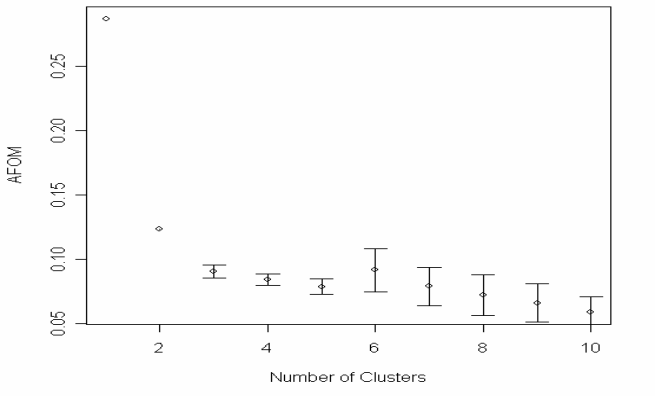

Fig. 2. AFOM graph for the Golub dataset.

TABLE V

NUMBER OF MISCLASSIFICATIONS FOR THE GOLUB DATASET – THREE CLUSTERS

Number of terminal nodes	2			4			6			R
Minimum terminal node size	1	5	10	1	5	10	1	5	10	R
AAMRT	7	7	6	3	3	3	7	9	8	7
	3	3	6	5	3	3	8	8	8	6
MRTPC	3	3	6	3	3	3	3	3	7	8
	6	6	6	5	3	5	3	3	9	8
MRTFS	7	7	18	5	5	17	3	5	16	3
	7	7	18	5	13	4	16	4	13	2

TABLE VI

IMPORTANT VARIABLES FOR THE GOLUB DATASET

Ensemble	Gene description
AAMRT	MB-1 gene
	LGALS1 Ubiquinol-cytochrome c reductase core protein II.
	PROBABLE PROTEIN DISULFIDE ISOMERASE ER-60 PRECURSOR
	DFD component of complement (adipsin)
	Zyxin
MRTPC	MB-1 gene
	Zyxin
	PROBABLE PROTEIN DISULFIDE ISOMERASE ER-60 PRECURSOR
	GLUTATHIONE S-TRANSFERASE, MICROSOMAL
	DFD component of complement (adipsin)
MRTFS	Growth factor receptor tyrosine kinase (STK-1) mRNA
	AFFX-M27830_5_at (endogenous control)
	GLYCOPHORIN B PRECURSOR
	CAPG Capping Protein (actin filament) gelsolin-like
	MDK Midkine (neurite growth-promoting factor 2)

VI. DISCUSSION

Firstly, we noticed no major trends with minimum terminal node size and number of terminal nodes. There may have been a very small effect of minimum terminal node size on the misclassification rate of SBK (three clusters) applied to the co-occurrence matrices of the Golub dataset. Because the smallest subgroup contains only nine observational units, if the minimum terminal node size was set too high (i.e. 10), this group could not be recovered perfectly.

The two criteria of SBK did not produce remarkably different results: criterion (9) could be performing slightly better than criterion (10). As the ultimate aim of SBK is prediction, it may be wise to employ the more commonly used squared error loss criterion.

Splitting the co-occurrence matrices of ensembles of AAMRTs and MRTPCs with SBK produced similar misclassification rates. The variable importance lists of these two ensembles were also alike. This indicates that the actual AAMRTs and MRTPCs were similar. The similarity between the results of AAMRTs and MRTPCs has been noted elsewhere [6]. The misclassifications using these two tree types were fairly stable. Furthermore, using random parameters gave a compromise misclassification rate of the ensembles grown using set parameters.

On the other hand, ensembles of MRTFSs, although capable of creating optimal solutions, tended to be fairly unstable and without any discernable pattern across minimum terminal node size and number of terminal nodes. A representative solution was not found by using random parameters.

The poor results obtained using MRTFSs were surprising. In a previous study these trees have been shown to outperform AAMRTs and MRTPCs on datasets perturbed by noise variables [6].

In the previous study the results of MRTPCs were generally stable to the number of factors. Here, we see that the stability of MRTPCs also extends to other parameters: the number of terminal nodes and minimum terminal node size.

The AFOM graphs indicated that there were three clusters within each dataset. The results were improved when the co-occurrence matrices of the Alon dataset were split to three clusters. Splitting the co-occurrence matrices of the Golub dataset unearthed the subgroups of the dataset. Generally, the misclassification rates agreed with other studies (see [25] and [19]). However, it is difficult to make a direct comparison because of different standardization (amongst other things).

The variable importance measures indicated similar genes across tree type. This was particularly evident with the important genes of ensembles of AAMRTs and MRTPCs. The genes deemed to be important by the algorithm agreed with those found in supervised classification studies. For example, the Zyxin gene of the Golub dataset is commonly selected in classification rules in [26]. In [22] Zyxin, MB-1, and adipsin are illustrated as genes useful in distinguishing AML from ALL. To highlight the power of the variable importance lists, we took the top five variables found by the ensemble of AAMRTs in Table VI and grew a single AAMRT using only these variables. For the two group case, the number of misclassifications decreased to four; and for the three group case, the number of misclassifications decreased to five. The variable importance lists here are derived without external knowledge of the grouping structure. Therefore, these important variables may determine not only known groups but also smaller subgroups. The important variables warrant further investigation as biomarkers.

Finally, because of the stability and dimension reduction associated with the ensembles of MRTPCs, we suggest using these trees to create the weighted co-occurrence matrices. If suitable parameters of the ensemble were unknown prior to analysis, it is advisable to use randomly selected values. With further research, weighted ensembles of MRTFSs could also give accurate clustering solutions. The optimal dimension of the response space deserves further investigation.

VII. CONCLUSION

Cluster analysis is an essential exploratory technique, often applied as a first step in the analysis of a large microarray dataset [27]. Cluster ensembles have been shown to give improved accuracy and stability over individual clustering solutions, in many fields [28],[29],[30] including Bioinformatics [19]. The improvements afforded by cluster ensembles on large datasets parallel results obtained with regression and classification ensembles. It is mooted that greater accuracy is attainable if the researcher is willing to

take a weighted aggregation of the individual clustering models to give the ensemble.

This research suggested a technique of creating a weighted cluster ensemble suitable for large datasets. Each cluster model, a multivariate regression tree, was weighted according to its predictive strength. The clusters were found in the response space of each tree; either the entire dataset, or the reduced dimension space constructed with the factor scores or the principal component scores of the dataset.

The resulting weighted co-occurrence matrix was split using SBK and the clusters agreed with the known groupings in the data. Interestingly, the technique uncovered two known subgroups in one dataset. Weighted co-occurrence matrices created with MRTPCs produced the most stable results across the datasets. Because of their stability and dimension reduction we recommend MRTPCs as the preferred tree type.

A valuable byproduct of the ensemble technique was an indication of the variables that were important in determining the clusters. Growing a single AAMRT on the variables selected as important, decreased the number of misclassifications. The important variables could warrant further investigation; some variables (genes) could be biomarkers of a disease.

REFERENCES

[1] L. Kaufman and P. Rousseeuw, "Clustering by means of medoids," in *Statistical Data Analysis based on the L_1 Norm*, Y. Dodge, Ed. Amsterdam: Elsevier, 1987, pp. 405-416.

[2] X. Z. Fern and C. E. Brodley, "Cluster ensembles for high dimensional data clustering: An empirical study," Oregon State University, Department of Computer Science, Technical Report CS06-30-02, 2006.

[3] C. Smyth and D. Coomans, "Predictive Weighting for Cluster Ensembles," *unpublished*.

[4] F. Questier, R. Put, D. Coomans, B. Walczak, and Y. V. Heyden, "The use of CART and multivariate regression trees for supervised and unsupervised feature selection," *Chemometrics and Intelligent Laboratory Systems*, vol. 76, pp. 45-54, 2005.

[5] C. Smyth, D. Coomans, Y. Everingham, and T. Hancock, "Auto-Associative Multivariate Regression Trees for Cluster Analysis," *Chemometrics and Intelligent Laboratory Systems*, vol. 80, pp. 120-129, 2006.

[6] C. Smyth, D. Coomans, and Y. Everingham, "Clustering noisy data in a reduced dimension space via multivariate regression trees," *Pattern Recognition*, vol. 39, pp. 424-431, 2006.

[7] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. New York: Springer-Verlag, 2001.

[8] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society Series B-Statistical Methodology*, vol. 58, pp. 267-288, 1996.

[9] T. Grotkjaer, O. Winther, B. Regenberg, and J. Nielsen, "Robust multi-scale clustering of large DNA microarray datasets with the consensus algorithm," *Bioinformatics*, vol. 22, pp. 58-67, 2006.

[10] K. Y. Yeung, D. R. Haynor, and W. L. Ruzzo, "Validating clustering for gene expression data," *Bioinformatics*, vol. 17, pp. 309-318, 2001.

[11] R. Tibshirani, G. Walther, D. Botstein, and P. Brown, "Cluster validation by prediction strength," *Journal of Computational and Graphical Statistics*, vol. 14, pp. 511-528, 2005.

[12] T. Hancock, "Multivariate Consensus Trees: Tree-based clustering and profiling for mixed data types," James Cook University, School of Mathematical and Physical Sciences, PhD Thesis 2006.

[13] L. Breiman, J. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*. Belmont, CA: Wadsworth, 1984.

[14] M. R. Segal, "Tree-Structured Methods for Longitudinal Data," *Journal of the American Statistical Association*, vol. 87, pp. 407-418, 1992.

[15] G. H. Dunteman, *Principal Components Analysis*. Newbury Park, CA: Sage, 1989.

[16] J. Friedman and B. Popescu, "Importance Sampled Learning Ensembles," Stanford University, Department of Statistics, Technical Report 2003.

[17] L. Kaufman and P. Rousseeuw, *Finding groups in data: an introduction to cluster analysis*. New York: Wiley, 1990.

[18] S. Monti, P. Tamayo, J. Mesirov, and T. Golub, "Consensus clustering: A resampling-based method for class discovery and visualization of gene expression microarray data," *Machine Learning*, vol. 52, pp. 91-118, 2003.

[19] S. Dudoit and J. Fridlyand, "Bagging to improve the accuracy of a clustering procedure," *Bioinformatics*, vol. 19, pp. 1090-1099, 2003.

[20] U. Alon, N. Barkai, D. A. Notterman, K. Gish, S. Ybarra, D. Mack, et al., "Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 96, pp. 6745-6750, 1999.

[21] E. Acuna and C. Rodriguez, "dprep: Data preprocessing and visualization functions for classification. R package version 1.0"

[22] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, et al., "Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring," *Science*, vol. 286, pp. 531-537, 1999.

[23] S. Dudoit and J. Fridlyand, "A prediction-based resampling method for estimating the number of clusters in a dataset," *Genome Biology*, vol. 3, pp. 0036.1-0036.21, 2002.

[24] R Development Core Team, "R: A language and environment for statistical computing" Vienna, Austria: R Foundation for Statistical Computing, 2006.

[25] G. J. McLachlan, R. W. Bean, and D. Peel, "A mixture model-based approach to the clustering of microarray expression data," *Bioinformatics*, vol. 18, pp. 413-422, 2002.

[26] S. G. Baker and B. S. Kramer, "Identifying genes that contribute most to good classification in microarrays," *BMC Bioinformatics*, vol. 7, 2006.

[27] S. Datta and S. Datta, "Methods for evaluating clustering algorithms for gene expression data using a reference set of functional classes," *BMC Bioinformatics*, vol. 7, 2006.

[28] D. Greene, A. Tsymbal, N. Bolshakova, and P. Cunningham, "Ensemble Clustering in Medical Diagnostics," presented at 17th IEEE Symposium on Computer-Based Medical Systems, Texas, 2004.

[29] A. Weingessel, E. Dimitriadou, and K. Hornik, "An Ensemble Method for Clustering," presented at Distributed Statistical Computing, Vienna, Austria, 2003.

[30] A. Strehl and J. Ghosh, "Cluster ensembles - a knowledge reuse framework for combining multiple partitions," *Journal of Machine Learning Research*, vol. 3, pp. 583-617, 2002.