Hybrid Hierarchical Clustering with Applications to Microarray Data

Hugh Chipman* and Robert Tibshirani†

August 30, 2005

Abstract

In this paper we propose a hybrid clustering method that combines the strengths of bottom-up hierarchical clustering with that of top-down clustering. The first method is good at identifying small clusters but not large ones; the strengths are reversed for the second method. The hybrid method is built on the new idea of a mutual cluster: a group of points closer to each other than to any other points. Theoretical connections between mutual clusters and bottom-up clustering methods are established, aiding in their interpretation and providing an algorithm for identification of mutual clusters. We illustrate the technique on simulated and real microarray datasets.

Key words: top-down clustering, bottom-up clustering, mutual cluster.

*Corresponding author. Hugh Chipman is Associate Professor and Canada Research Chair in Mathematical Modelling, Department of Mathematics and Statistics, Acadia University.

†Robert Tibshirani is Professor of Health Research and Policy, and Statistics, Stanford University.

Chipman was partially supported by funding from NSERC, MITACS, and CFI. Tibshirani was partially supported by NIH grant 2 R01 CA72028, and NSF grant DMS-9971405.

1 Introduction

Agglomerative (bottom-up) hierarchical clustering algorithms are important analysis tools for microarray data, They are useful for organizing the genes and samples from a set of microarray experiments so as to reveal biologically interesting patterns. Hierarchical methods are especially useful, because they enable analysts to simultaneously examine groupings of the data into many small clusters (e.g. repeat samples from the same patient) and a few large clusters (e.g. different prognosis groups). The use of a single hierarchical model ensures that groups at different levels of detail are nested, facilitating interpretation.

Bottom-up clustering, which successively joins objects, is good at identifying small clusters, but can provide sub-optimal performance for identifying a few large clusters. Conversely, top-down methods are good at identifying a few large clusters, but weaker at many small clusters. We seek to combine the strengths of both approaches, modifying top-down procedures with information gained from a preliminary bottom-up clustering. A new key concept is that of a *mutual cluster*, defined as a group of objects collectively closer to each other than to any other object. In this paper we illustrate how this technique and others can be used to help interpret existing bottom-up clusterings, and to produce more effective hybrid hierarchical clusterings.

Section 2 defines a mutual cluster. Section 3 briefly reviews clustering methods, and discusses top-down and bottom up methods. Section 4 proposes a hybrid method that uses mutual clusters. The three methods are compared, with simulation experiments (Section 5) and a microarray example (Section 6). The paper concludes with a discussion.

2 Mutual Clusters

We introduce the concept of a mutual cluster as a group of points which are sufficiently close to each other and distant from all other points that they should never be separated. Formally we define a *mutual cluster* to be a subset S of the data, such that with distance function d,

$$\forall x \in S, y \notin S, d(x,y) > \text{diameter}(S) \equiv \max_{w \in S, z \in S} d(w,z). \tag{2.1}$$

That is, we require that the largest distance between points in S be smaller than the smallest distance from a point in S to any point not in S. Figure 1 illustrates this concept in two dimensions using Euclidean distance. Observations 1,2,3 form a mutual cluster. The diameter of the mutual cluster is the largest distance between points in the mutual cluster, here the segment d_{13}. A circle of radius d_{13} is drawn around each of $\{1,2,3\}$. All other points lie outside these circles, so all distances between $\{1,2,3\}$ and $\{4,5,6,7,8\}$ are greater than the diameter, making $\{1,2,3\}$ a mutual cluster. If a 9th point were added and it fell in a circle, $\{1,2,3\}$ would no longer be a mutual cluster. Depending on the location of the new point, the set $\{1,2,3,9\}$ might become a mutual cluster.

By definition, the entire dataset is a mutual cluster. Moreover, a single distant outlier could make the remaining points a mutual cluster. As this is an exploratory technique, if such anomalies are present in a dataset, they are almost certain to be detected. In fact, a mutual cluster containing all but a few observations could flag those observations as outliers.

Before discussing algorithms for the identification of mutual clusters, we note the following property. The result and the rest of the section requires some familiarity with bottom-up (agglomerative) hierarchical clustering. Unfamiliar readers will find a review in Section 3.

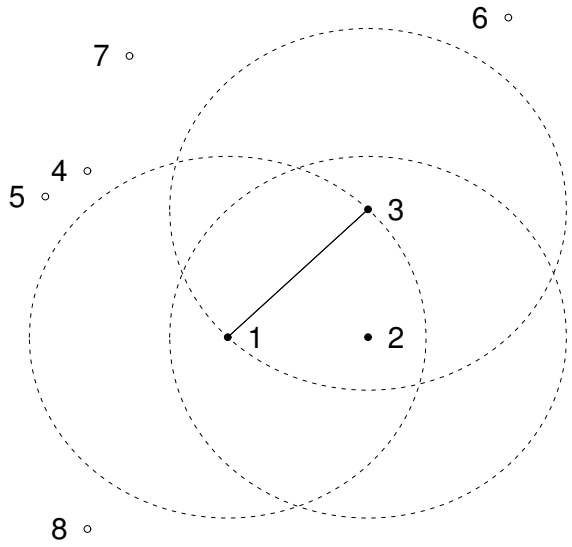

Figure 1: *Mutual cluster illustration. Points 1,2,3 form a mutual cluster since all other points are more than d_{13} away from points 1,2 and 3. The largest within-cluster distance (d_{13}) is the cluster diameter.*

Theorem: *A mutual cluster is not broken by bottom-up clustering with any of single, average, or complete linkage.*

A proof is given in the appendix.

This theorem has several significant implications. First, it implies that mutual clusters can be identified by examining nested clusters from bottom-up clustering. Each nested cluster for which (2.1) holds is a mutual cluster. Since mutual clusters cannot be broken by bottom-up methods, checking all nested clusters guarantees that all mutual clusters will be found.

A second implication is that this theorem aids in the interpretation of bottom-up clustering methods. In particular, graphical displays of bottom-up clusterings such as dendrograms can be augmented to include identification of mutual clusters. Figure 4 presents an example, in which mutual clusters are indicated by a hexagon near their joining point in the dendrogram. This additional information can aid in the interpretation of mutual clusters, or in the decision of what clusters to divide. This is explored further in Section 6.2, including an interesting connection with common practice in the analysis of microarray data.

Lastly, the theorem adds further support to the idea that mutual clusters contain strong clustering information: No matter which of single, complete or mean linkage is used in bottom-up clustering, mutual clusters are not broken.

3 Review of clustering methods

Before showing in the next section how mutual clusters can be used to construct a hybrid clustering algorithm, we provide a brief review of clustering methods. Additional details may be found in books that review clustering, such as Gordon (1999), Chipman, Hastie & Tibshirani (2003) (for emphasis on microarray data),

or chapter 14 of Hastie, Tibshirani & Friedman (2001).

Clustering methods seek to divide observations into similar groups, typically in a "hard" fashion in which each observation is assigned to a single group. Soft methods, such as mixture models, are not considered here. The notion of similarity of observations requires that a distance or dissimilarity measure between two observations be defined. Euclidean distance is the most common. For microarray data, the correlation between two vectors can also be used. This is equivalent to the Euclidean distance between the two vectors after each is normalized to have mean 0 and standard deviation 1.

Many popular clustering can be characterized as either partitioning methods, which seek to optimally divide objects into a fixed number of clusters, or hierarchical methods, which produce a nested sequence of clusters.

The K-means algorithm (Lloyd 1957) is the most popular of partitioning algorithms. It seeks to find K clusters that minimize the sum of squared Euclidean distances between each observation and its respective cluster mean. In its simplest form, the K-means algorithm iteratively alternates between two steps: (1) for a given set of cluster centers, assign each observation to the cluster with the nearest center, and (2) for a given assignment of observations to clusters, update each cluster center as the sample mean of all points in that cluster. Initial center values for step 1 are often a random sample of K observations. It typically converges to one of the many local optima, rather than the global optimum. Hartigan & Wong (1979) give a more complicated algorithm which is more likely to find a good local optimum. Whatever algorithm is used, it is advisable to repeatedly start the algorithm with different initial values, increasing the chance that a good local optimum is found.

Other partitioning methods have been developed, including K-medoids (Kauf-

man & Rousseeuw 1990), which is similar to K-means but constrains each cluster center to be one of the observed data points. Self-organizing maps (Kohonen 1989) also have similarities to K-means, but arrange clusters on a lattice to aid interpretation. Partitioning methods are not used here, except as a building block in some hierarchical algorithms.

As mentioned in the introduction, hierarchical clusterings of the data may be constructed by either bottom-up methods, which successively join nearest points and groups of points, or by top-down methods, which successively divide up groups of points into subgroups. Many hierarchical clustering algorithms have an appealing property that the nested sequence of clusters can be graphically represented with a tree, called a *dendrogram*. Figure 2 presents an example, giving 6 points in two dimensions (left side) and the corresponding dendrogram (right). The nested clusters, determined in this case by a bottom-up algorithm, are indicated in the left plot of Figure 2 by nested polygons. The height at which groups of points are joined corresponds to the distance between the groups. Thus observations 1 and 6 are considerably closer than observation 5 is to the group {1,3,4,6}.

In order for the bottom-up method to decide what groups to join, a distance or dissimilarity between two groups must be defined. Four popular methods of measuring distance between groups are:

1. Single linkage, which uses the minimum distance between points in different groups,
2. Complete linkage, which uses the maximum distance,
3. Mean linkage, which uses the average of all distances between points in the two groups,
4. Centroid linkage, which uses the distances between group centroids (e.g. group

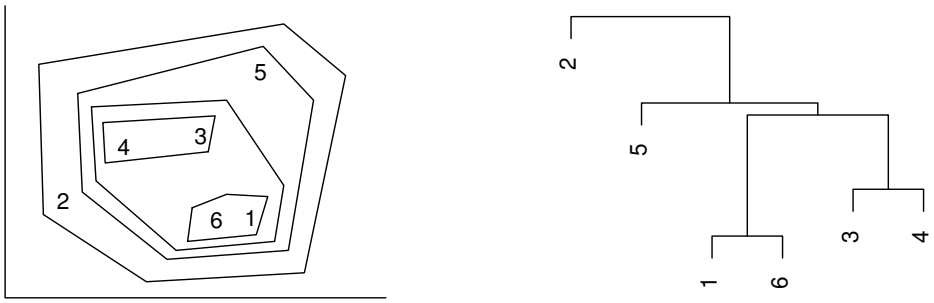

Figure 2: *Simple example of hierarchical clustering.*

means).

Each method will result in a different clustering. Single linkage tends to produce long chains of points. Complete linkage tends to produce compact, spherical clusters. Mean and centroid linkages represent compromises between the two extremes. In the analysis of microarray data, Eisen, Spellman, Brown & Botstein (1998) popularized a rediscovered version of bottom-up centroid linkage clustering. Other variations exist, such as Ward's method, which joins clusters so as to minimize the within-cluster variance.

For top-down methods, the successive division of groups can be accomplished by a variety of algorithms. Each division is a two-group partitioning problem. Thus recursive application of K-means with $K = 2$ is one approach to top-down clustering. This is known as *tree-structured vector quantization* (TSVQ) (Gersho & Gray 1992) in the engineering literature, where $K-$means is referred to as vector quantization. Our particular TSVQ implementation uses 20 attempts of 2-means at

each partitioning step, to find a good partition of the data. Multiple attempts with random starting values significantly improve the quality of clusters identified. Other methods of recursive subdivision are possible. For example, Macnaughton-Smith, Williams, Dale & Mockett (1965) identifies the single most dissimilar observation as a new cluster, and successively moves to this cluster all observations closer to it.

One important consideration in the use of clustering methods is missing values. Many bottom-up methods and that of Macnaughton-Smith et al. (1965) are able to handle missing values. K- means (and thus TSVQ) require complete data, requiring imputation before clustering.

4 Hybrid hierarchical clustering using mutual clusters

In addition to providing useful information, mutual clusters may be used as a building block in other clustering algorithms. In this section, we develop a hybrid hierarchical method that makes use of multiple clusters.

Since hierarchical methods are the focus of this paper, we present a simple motivating example. Figure 3 illustrates the results of bottom-up, top-down and a hybrid clustering of the data earlier presented in Figure 2. There are two mutual clusters: $\{3,4\}$ and $\{1,6\}$. The hierarchical clusterings are indicated by nested polygons. If two clusters are required, we get $\{2\},\{1,3,4,5,6\}$ (bottom-up) and $\{2,4\},\{1,3,5,6\}$ (top-down). Top-down clustering seems superior to bottom-up when a small number of clusters are required. Conversely, if a large number of clusters are required, top-down is inferior, since it separates 3 and 4, which are a mutual cluster.

The concept of mutual clusters can be used to create a hybrid method. This

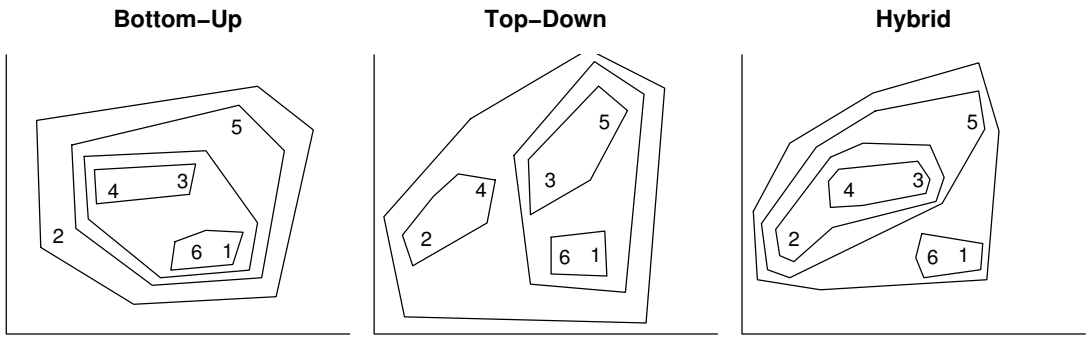

Figure 3: *Simple example to illustrate different clustering methods. Six points in two dimensions are hierarchically clustered using three different methods. Polygons are used to indicate nested clusters.*

method has three steps:

1. Compute the mutual clusters.
2. Perform a constrained top-down clustering (via TSVQ), in which each mutual cluster must stay intact. This is accomplished by temporarily replacing a mutual cluster of points by their centroid.
3. Once the top-down clustering is complete, divide each mutual cluster by performing a top-down clustering *within* each mutual cluster.

By treating each mutual cluster as an indivisible object, we avoid top-down's tendency to break up groups that should be kept together. Figure 3 (right) illustrates this: the hybrid method retains the mutual clusters $\{3,4\}, \{1,6\}$ while generating a more appealing clustering of the data into two groups: $\{2,3,4,5\}$ and $\{1,6\}$.

The modular nature of the hybrid approach enables many variations. In step 2, any top-down method could be employed. In step 3, an alternative would be to employ a bottom-up algorithm. Since mutual clusters are often small, the use of either top-down or bottom-up in step 3 will give similar results within mutual clus-

ters. Using top-down at both stages seems simpler: only one clustering algorithm is used, and the same objective function is used at both stages (sometimes bottom-up and top-down methods use different objective functions).

The next two sections compare the three methods (bottom-up, top-down, hybrid), first in simulation experiments (Section 4) and then applied to a microarray dataset (Section 5).

5 Simulation experiments

To illustrate hybrid clustering, we consider four different simulations, involving 200 observations in 500 or 10,000 dimensions. 100 replicates of each simulation will be carried out, with several cluster characteristics varying within replications. In all simulations, some clusters will be chosen to be mutual clusters. Performance of top-down (TSVQ), bottom-up (mean linkage agglomerative), and hybrid clustering will be assessed in terms of (i) retention of mutual cluster structure, (ii) correct classification of points to clusters, and (iii) within-cluster sums of squares. We will view the hybrid method as successful if it can retain mutual cluster structure (i) while at the same time matching the ability of top-down methods to find good partitions (ii and iii).

5.1 Simulation 1: 50 small clusters

In this simulation, 50 clusters with 4 observations each are generated. Initially, observation i in cluster j is generated as

$$x_{ij} = \mu_i + \epsilon_{ij}, \qquad \mu_i \sim N(0, I), \qquad \epsilon_{ij} \sim N(0, 3^2 I). \tag{5.1}$$

Although variability is larger within clusters than between, the fact that 500 variables carry the "signal" implies there is still sufficient structure that the clusters

Experiment	dimension	# clusters	# mutual clusters	mean cluster size	sd of cluster size
1	500	50.00	12.82	4.00	0
2	500	34.84	12.67	5.78	3.97
3	500	9.39	2.88	21.50	18.36
4	10,000	34.84	3.53	5.78	3.97

Table 1: *Summaries of cluster structure for four simulation experiments. All values are averages over 100 replications of each experiment.*

can be identified. Clusters are then randomly selected (with probability 0.5) as candidates to become mutual clusters. To convert a mutual cluster candidate into a mutual cluster, the observations are shrunk toward cluster centroid $\hat{\mu}$ by a random quantity Bt. Shrunken observation $\tilde{x}_i$ is given by $\tilde{x}_i = Btx_i + (1 - Bt)\hat{\mu}$, with $0 < B < 1$. The upper threshold t is the largest value that could be used when $B = 1$ and still yield a mutual cluster. The random variable $B \sim$Beta(500,2), giving $\Pr(B < 0.99) = 0.04$. Thus, the resulting mutual clusters are just slightly smaller than the MC threshold. The mutual clusters are generated sequentially in the fashion described above. This sequential generation implies that subsequent shrinkage of different clusters may invalidate earlier constructed mutual clusters. For this simulation, roughly 25% of clusters are mutual clusters.

The characteristics of this data over the 100 replications are summarized in Table 1.

Three measures of performance are used for each clustering method: the number of "broken" mutual clusters, the percentage of misclassified points, and the (relative) within-cluster sum of squares. A hierarchical clustering of points "breaks" a mutual

cluster if any points outside the mutual cluster are merged with some of the MC points before all MC points are merged. Put another way, there is no possible division of the dendrogram into clusters that groups together all points in the mutual cluster and no other points. In our study, the top-down method is the only one that can break mutual clusters, since by construction, the hybrid method preserves this structure, and as proven earlier in Section 2, bottom-up methods cannot separate mutual clusters. Only mutual clusters that are present in the data by construction are considered in the performance measure.

Both of the remaining performance measures (misclassification and WSS) are calculated by first dividing each hierarchical clustering into the true number of clusters (50 in this simulation). Misclassification is then calculated by examining each pair of points and considering whether they are together or apart in the true and estimated clustering. The points will be misclassified if they are together in one clustering and separate in the other clustering. Over all pairs, this yields a misclassification measure

$$M(C,T) = \frac{\sum_{i>i'} |I_C(i,i') - I_T(i,i')|}{\binom{n}{2}}. \tag{5.2}$$

where C is a partition of points into clusters and $I_C(i,i')$ is an indicator for whether clustering C places points i and i' in the same cluster. I_T is similarly defined for the partition T corresponding to the true clustering. $M(C,T)$ is proportional to the Hamming distance between the two indicators, and was used in the context of decision tree partitions by Chipman, George & McCulloch (1998). The denominator scales the measure so 0 indicates perfect agreement and 1 indicates total disagreement.

The WSS measure is calculated on the basis of partitioning the dendrogram into

Experiment	% MC broken by TD	Misclassification			Relative WSS		
		H	BU	TD	H	BU	TD
1	28.1%	1.4%	41.0%	1.7%	1.02	1.09	1.03
2	7.1%	2.2%	64.0%	2.2%	1.02	1.13	1.02
3	1.3%	6.9%	64.1%	7.3%	1.03	1.24	1.04
4	18.0%	3.1%	65.2%	3.0%	1.00	1.02	1.00

Table 2: *Performance of clustering methods (H=Hybrid, BU = Bottom-up, TD = Top-down) on simulated datasets. See text for definitions of broken points, misclassification, and relative WSS. For all three measures, good performance is indicated by a small value. Relative WSS is a ratio.*

the true number of clusters. Within-cluster sum of squared distances is

$$WSS = \sum_{k=1}^{K} \sum_{C(i)=k} ||x_i - \hat{\mu}||^2 \tag{5.3}$$

In our comparisons, relative WSS is reported, as the ratio of WSS for the estimated clustering to WSS for the true clustering. Thus a relative WSS of 1.0 indicates the same WSS as the true clustering.

Performance results for the first simulation are presented in the first row of Table 2. Over a quarter of mutual clusters are broken by the top-down method (and of course none are broken by bottom-up or hybrid). The hybrid and top-down methods have very low misclassification rates, while bottom-up misclassifies over 40% of pairs. A possible explanation is that the bottom-up method has many singleton observations that are joined near the top of the dendrogram. Differences in WSS are smaller between the three methods, although again bottom-up is worst.

5.2 Simulation 2: small clusters of random size

The mechanism generating data is similar to the first simulation, except

- The 200 observations are randomly divided into a random number of clusters, each of random size. This is accomplished by generating a $D \sim Dirichlet(n = 50, a = (1, 1, \ldots, 1))$. The number of observations in each of 50 clusters is set by rounding $200D$, and eliminating all components with 0 or 1 observations. This yields (see summaries in Table 1) an average of 38.84 clusters, with a mean of 5.78 observations per cluster.
- The shrinkage factor Bt for generation of mutual clusters is now drawn as $B \sim Beta(20, 2)$, implying $\Pr(B < 0.90) = 0.36$. Recall that $B < 1$ will yield a mutual cluster, so some mutual clusters will be more tightly clustered ($B \ll 1$) than in the previous example.

As indicated in Table 2, top-down methods still break a significant portion of mutual clusters (7.1% on average). The hybrid method has performance similar to the top-down method, and the results resemble the first simulation. Misclassification performance of bottom-up is considerably poorer than in the first example, and relative WSS values for bottom-up are also degrading.

5.3 Simulation 3: larger clusters of random size

The general mechanism for data generation is the same as in the second example. Parameters are varied so as to generate larger clusters, and mutual clusters that are more likely to be near the existence boundary ($B < 1$ but close to 1). In particular, $D \sim Dirichlet(n = 10, a = (1, \ldots, 1))$ and $B \sim Beta(500, 2)$. This yields an average of 9.39 clusters and 2.88 mutual clusters. Average cluster size is 21.50 observations, with considerable variation. Although 50% of clusters are

meant to be mutual clusters, sequential shrinkage invalidates more clusters because of larger cluster sizes.

In this scenario, only a small number of mutual clusters are broken. Misclassification rates are worse than in the other two simulations, although still quite good for top-down and hybrid methods. Relative WSS values are similar for hybrid and bottom-up, and quite a bit worse (1.24) for bottom-up clustering.

5.4 Simulation 4: high dimension

The data generation mechanism is the same as simulation 2, with two exceptions. The 500 dimensional observations in simulation 2 are augmented by including an additional 9,500 dimensions of "junk" variables, each distributed as $N(0, 1.5^2 I)$. Mutual clusters are generated using $B \sim Beta(80, 20)$, implying $\Pr(B < .8) = 0.48$. Top-down methods continue to break a considerable number of mutual clusters (18%), while bottom-up methods suffer from misclassification problems.

Other experiments with higher-dimensional data indicate that when the cluster signal is present in all dimensions (as opposed to 500 out of 10,000 dimensions here), mutual clusters are unlikely to be broken by the top-down algorithm unless within-cluster variance is very large.

5.5 Summary

The weaknesses of top-down and bottom-up methods are illustrated in these examples. Top-down methods can break mutual clusters, while bottom-up methods can suffer from poor partitioning of observations into the true number of clusters. The hybrid approach seems to gain the best of both worlds, not breaking mutual clusters, while retaining superior partitioning when the correct number of clusters is used. The tendency of top-down clustering to break mutual clusters seems great-

est when there are a large number of clusters. A top-down method will have to make more binary splits to reach the true number of clusters in such a situation, increasing the chances that at least one of the splits breaks a mutual cluster.

Our particular choice of competitors (TSVQ and mean linkage agglomerative) enables reasonable comparisons to be made with the hybrid method. In particular, TSVQ is employed in the hybrid method, and the mean linkage guarantees that mutual clusters will not be broken. Other bottom-up linkage methods, such as Ward's method (which seeks to minimize variance at joins) may break mutual clusters, and consequently are not considered.

6 Breast Cancer Tumors

The data are described in Sorlie et al. (2001) which also use clustering to identify tumor subclasses. For each of 85 tissue samples, 456 cDNA gene expression values were used. Approximately 5.6% of data values are missing. Before any clustering was carried out, the missing values were imputed using a k-nearest neighbours strategy (see Chipman et al. (2003), Section 4.2.1 and references therein for details). All clustering illustrated in this section focuses on groupings of observations (i.e., tissue samples).

6.1 Mutual Clusters

We begin by identifying mutual clusters. The correlation distance between samples is used. This is equivalent to squared Euclidean distance after scaling. For the 85 subjects, there are 17 mutual clusters, 16 of which contain just two points. These mutual clusters and their diameters are listed in Table 3, along with the distance to the nearest point not in the mutual cluster, and the number of distances in the

id	diameter	distance to nearest outsider	# smaller distances in data	broken by top-down
31 32 33 34	0.2992*	0.3715	0	
29 30	0.3366	0.3715	6	
7 8	0.4090	0.4733	8	
35 36	0.4342	0.5092	10	
10 12	0.4579	0.4848	15	
24 25	0.4642	0.6229	16	
63 64	0.5238	0.5960	34	
65 66	0.5542	0.5693	41	×
82 83	0.5661	0.5794	46	
16 20	0.5953	0.6525	61	
50 51	0.6134	0.7153	72	
37 60	0.6275	0.6417	80	×
78 79	0.6416	0.6556	92	
43 44	0.6573	0.6675	104	×
45 46	0.6723	0.6911	117	×
52 55	0.7043	0.7308	155	
54 56	0.7610	0.7683	277	×

* distances between 31,32,33,34 below:

	31	32	33
32	0.2992227		
33	0.1934151	0.2052296	
34	0.2491986	0.2662368	0.151286

Table 3: *Mutual clusters for Sorlie data, ordered by cluster diameter. In all cases but* $\{31, 32, 33, 34\}$*, the diameter is the only distance in the mutual cluster. The "broken by" column identifies mutual clusters that are broken by top-down algorithms.*

dataset smaller than the diameter (out of a total of 3570 distances between pairs). The distances within the mutual cluster {31, 32, 33, 34} are also given. These four points are the only normal tissue samples in the dataset, so it is encouraging that they form a mutual cluster. Two of the three benign tissue samples (35, 36) also form a mutual cluster.

The diameters of the mutual clusters vary widely, from 0.2992 to 0.7610, the latter being more than half the largest distance in the entire dataset. For a set of points with a large diameter to be a mutual cluster, they must be quite distant from

all other points. This is seen in the third column, where the nearest outsider to a mutual cluster is often just slightly more distant than the diameter. The column giving the number of smaller distances in the data reveals that a number of close points are not mutual clusters, since the ranks exclude many values. Such points do not form mutual clusters because addition of nearby points to the group would result in a sufficiently large diameter so as to include other points. Evidently, a mutual cluster can be defined by very small distances, and/or a set of points that are far away from all other points in the dataset.

6.2 Hierarchical clustering of the data

As in Section 5, the bottom-up method utilizes mean linkage, the top-down method is TSVQ, and the hybrid method based on TSVQ described in Section 4 is used. All methods use the same distance measure, since K-means in TSVQ utilizes squared Euclidean distance, which is equivalent to the correlation distance used here in the bottom-up algorithm. Top-down and bottom-up clustering trees are given in Figure 4. The horizontal lengths of the dendrogram branches cannot be compared between plots, as the bottom-up clustering is measuring average squared Euclidean distance between clusters, while the top down method uses sum of squared Euclidean distance between all points in different clusters. The vertical axes are used primarily to gauge the order in which clusters are joined.

The top-down algorithm breaks five mutual clusters. Table 3 and Figure 4 (a) identify the broken mutual clusters (last column of the table, and by labels such as `*** A ***` in the plots). In Figure 4 (a), points belonging to the same broken mutual cluster are identified by the same letter. In some cases (e.g. $\{65, 66\}$) the mutual clusters are quite close yet they are still broken. Worse still, points that belong to a broken mutual cluster can be separated by a large number of divisions.

Points 37 and 60 are not merged until the last join in the dendrogram.

The mutual clusters are indicated by diamonds in the bottom-up dendrogram (Figure 4 (b)). Although dendrograms are usually cut at a certain height to produce clusters, there is no single cut that can be made that will separate each mutual cluster from all other data points. Mutual clusters appear to employ a more "local" definition of similarity, grouping more distant points together in regions of the space where the nearest other neighbours are distant. An interesting connection with the analysis of microarray data is that researchers also choose partitions that correspond to no single cut of a dendrogram, perhaps also employing the notion of local similarity.

Before further comparing the clustering methods, we give the hybrid clustering of the data, shown in Figure 5. No mutual clusters are broken (by definition of the algorithm). All mutual clusters are labeled in this plot, along with both the numeric labels of points and the longer descriptive labels.

The remainder of this section considers a number of comparisons of the three clustering methods. A fourth clustering is also compared, that of Sorlie et al. (2001). In that paper, Eisen's centroid method was applied to the data containing missing values. The final clustering identified five groups. This clustering does not correspond exactly to a horizontal slicing of the dendrogram; some points were assigned to different clusters heuristically. For comparison to this clustering, we will divide each of our hierarchical clusterings into five clusters.

The first comparison with the five Sorlie clusters continues the idea of broken mutual clusters. Any particular partition of the data into clusters will "break" a mutual cluster if points in the mutual cluster do not fall into the same partition of the data. The five Sorlie clusters break 2 mutual clusters, while the top-down method breaks 4 mutual clusters. By definition, bottom-up and hybrid methods

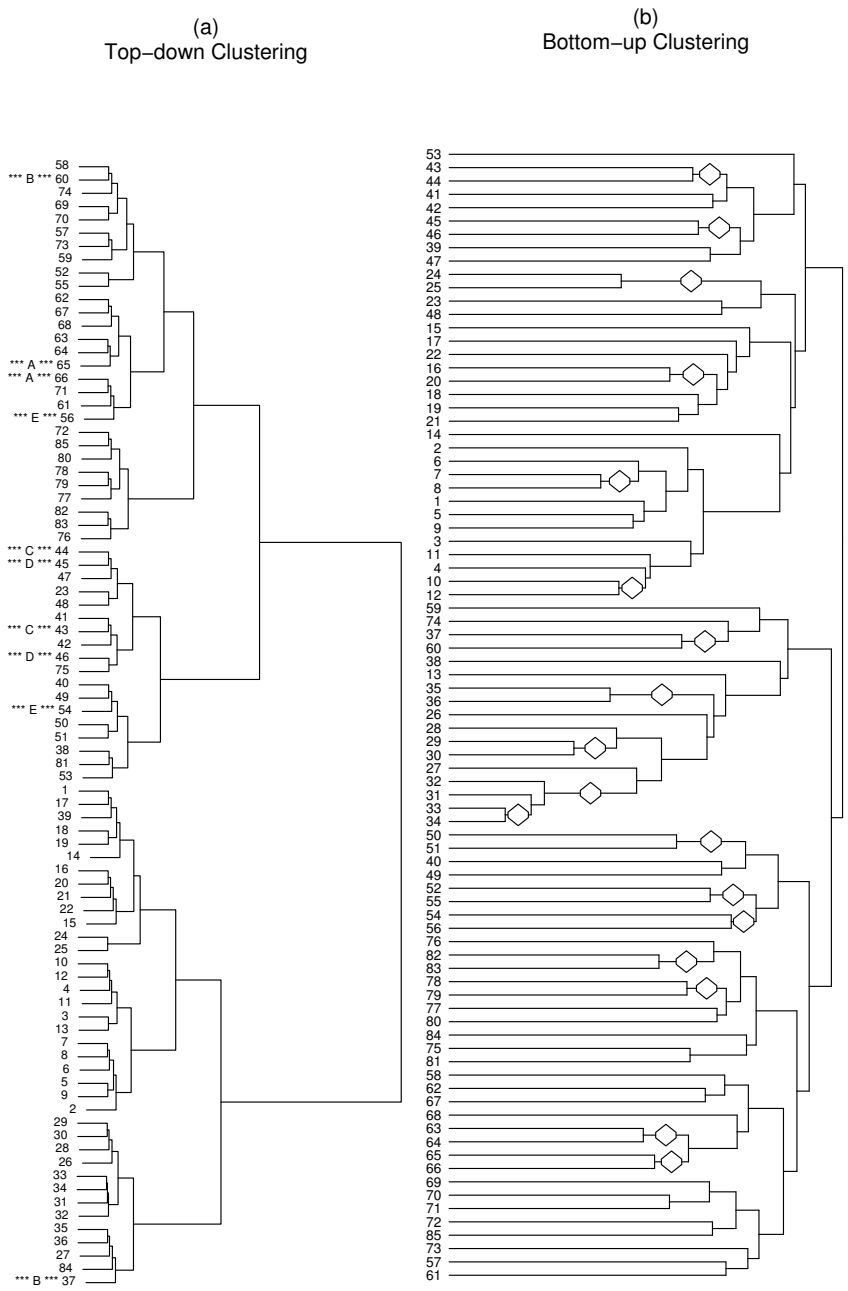

Figure 4: *Clusterings of 85 observations, using top down (a) and bottom-up (b) methods. Mutual clusters are indicated by dendrogram labels in (a) and hexagons in (b). The top down method breaks several mutual clusters. The branch lengths are not comparable between the two dendrograms, as discussed in the text.*

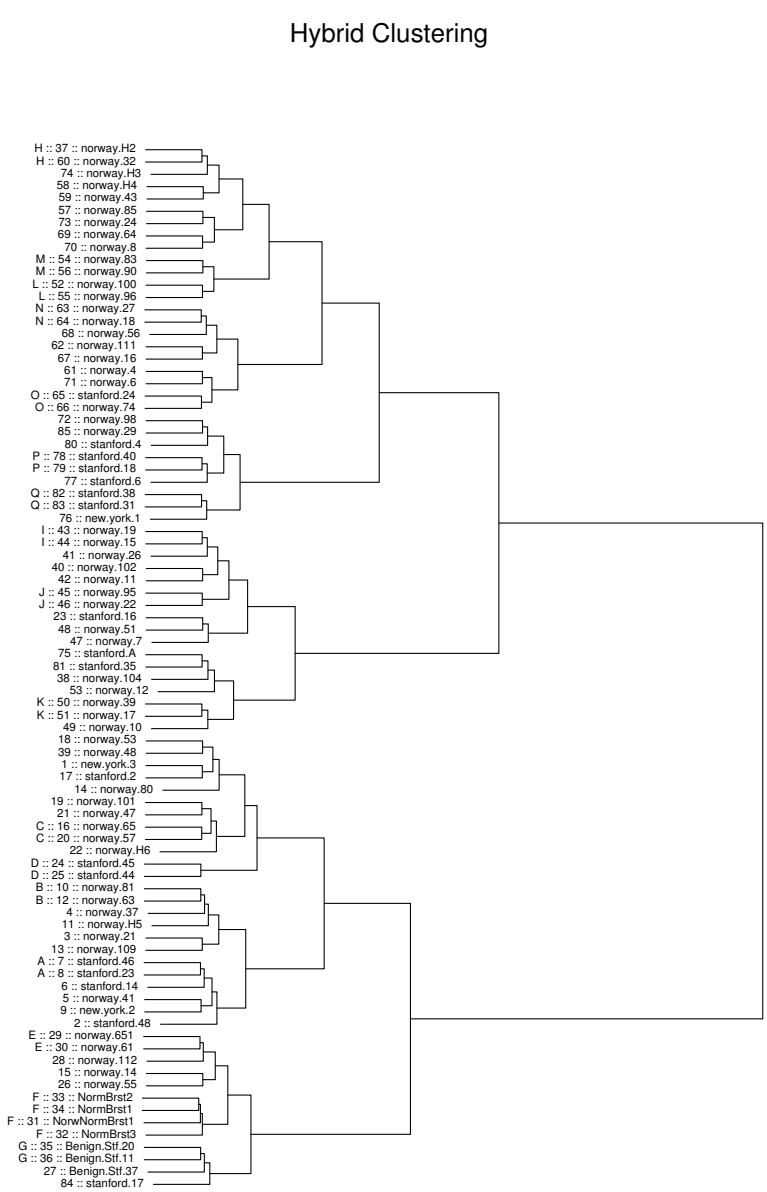

Figure 5: *Clusterings of 85 observations, with mutual clusters forced to remain together.*

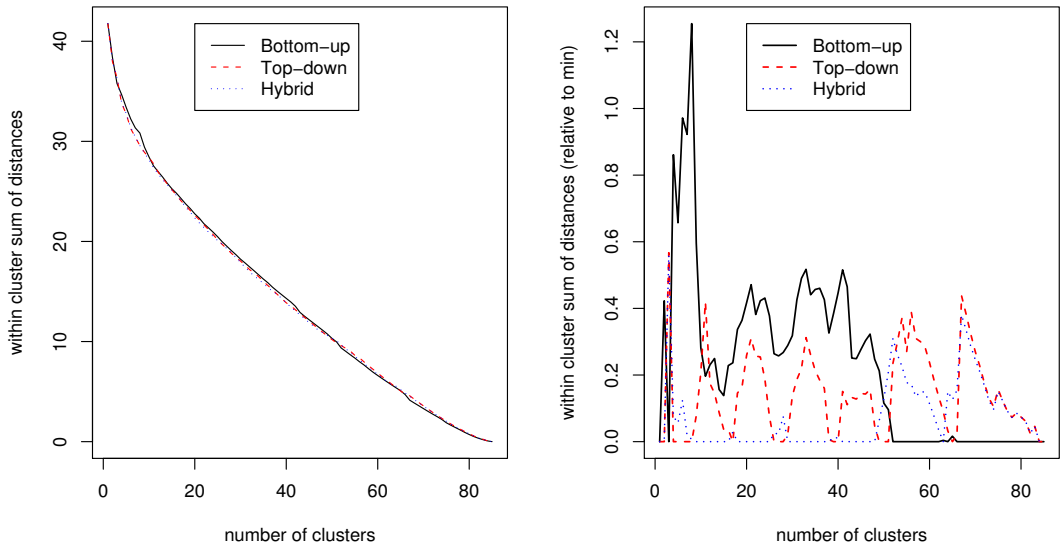

Figure 6: *Within-cluster sum of distances for the three clusterings (left) using (5.3). The smallest sum is subtracted off for each cluster size in the right plot.*

don't break any mutual clusters. The number of broken mutual clusters differs when using a partition instead of the full tree structure because a mutual cluster could be broken but still have all points in the same partition.

Another performance measure for the clusters is the ability to identify data partitions that have minimum within-partition sum of (squared) distances. This is assessed in Figure 6 for the three hierarchical methods. A nested sequence of partitions is generated by each clustering method, by cutting the corresponding dendrogram at an appropriate height. For each number of partitions, the sum of all pairwise within-group (squared Euclidean) distances (5.3) is calculated. Good clusterings will minimize this sum. The sum of distances is plotted as a function of number of clusters (left plot). As all clusters are quite close, differences are accentuated in the right plot, where the distance relative to the minimum is plotted.

In the plot we see that for a small number of clusters, the top-down method does well. This is to be expected, since top-down begins optimization with one cluster. Bottom-up does best only when there are a large number of clusters. The hybrid method tracks the top-down method most closely, with optimal or near-optimal performance for 1-50 clusters. In this example, the top-down stage of the hybrid method is not affected much by the mutual cluster constraint, and this constraint has little impact at later stages of the top-down clustering.

One final comparison indicates the degree of similarity between the different hierarchical methods. The pairwise misclassification measure $M(C,T)$ in (5.2) was presented to compare an estimated partition C to a true partition T. The measure is symmetric in C and T, and if the two partitions are arbitrary, $M(C,T)$ provides a measure of disagreement between the cluster.

The dissimilarities $M(C,T)$ between dendrogram are plotted in Figure 7 as a function of the number of partitions. The top-down and hybrid methods are quite similar for all values of K. Differences between either of these methods and bottom-up are quite large for a few clusters, and diminish as the number of clusters approaches n. At n clusters all methods must agree since there is only one possible partition.

The three methods can also be compared to the Sorlie clustering (with 5 clusters). Distance measures are as follows: Bottom-up to Sorlie: 0.133, Top-down to Sorlie: 0.169, Hybrid to Sorlie: 0.170. As expected, the Sorlie clustering is most similar to the bottom-up clustering.

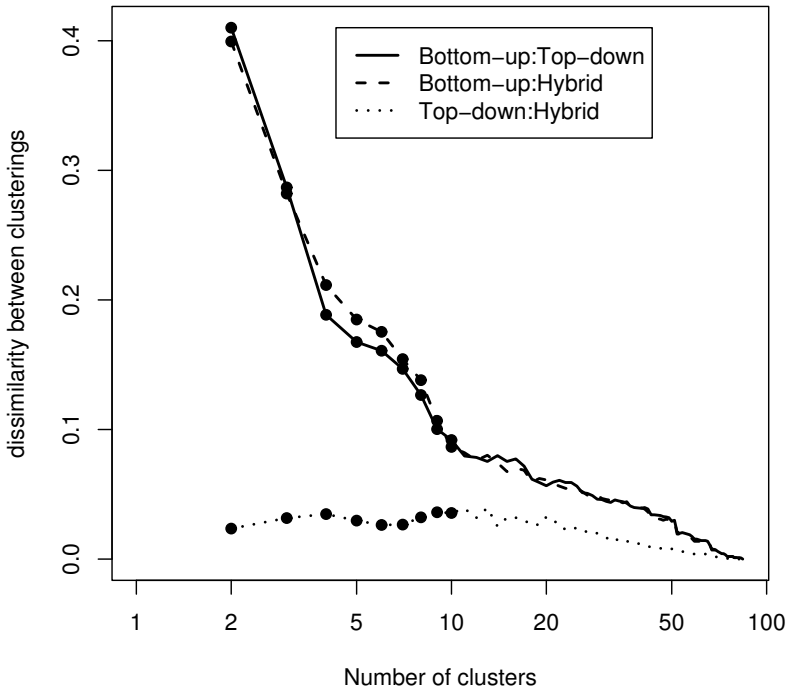

Figure 7: *Comparison of dissimilarity between three clusterings, using (5.2). The distance between cluster partitions is plotted as a function of number of partitions.*

7 Discussion

We have proposed a hybrid clustering method that uses the concept of mutual clusters. In simulated and real data examples, it offers advantages over both bottom-up and top-down clustering. Mutual cluster structure is preserved by the hybrid algorithm, which also retains the ability of top-down methods to accurately divide data into good clusters.

The fact that mutual clusters are defined by a precise criterion can be both a strength and weakness. The advantage of such a definition is its intuitive nature. A possible disadvantage is sensitivity to data variation. Several mutual clusters in Table 3 have diameters just smaller than the distance to the next closest point. A small movement of this next closest point could either

- cause it to join and form a larger mutual cluster, or
- prevent any of the points from being a mutual cluster.

Points 65 and 66 are a good example: their diameter is 0.5542, and the nearest point is a distance of 0.5693, just slightly larger than the diameter.

Another related issue is the limiting behavior of mutual clusters as sample size increases. With increasing density of data points, any set of points is increasingly likely to have other points within its mutual cluster boundary, making it less likely to be a mutual cluster. It should be noted that this problem is most acute in low dimensions, which "fill up" faster than high dimensional spaces. For microarray data, dimensionality is very high, making this concern less of an issue. This is seen in the number of large diameters in Table 3, some of which are about half the largest inter-point distance in the dataset. If the data were dense, mutual clusters would be unlikely to have such large diameters.

The definition of a mutual cluster might be generalized, so as to deal with these

issues. For example, in Figure 1, spheres of radius αd_{13} with $0 < \alpha < 1$ could be used around points 1, 2, 3, rather than radius d_{13}. Another possibility would be "soft" definition of a mutual cluster, allowing a small proportion of points outside the mutual cluster to fall within the mutual cluster boundaries.

R code for the examples is available from the first author's webpage.

Appendix: Proof of Theorem

Proof (by contradiction): Consider complete-linkage bottom-up clustering, in which the distance between two groups of points is the maximum of all pairwise inter-group distances.

Let S be a mutual cluster, and suppose that $l \notin S$ is a point that is joined to a strict subset $P \subset S$. That is, l breaks the mutual cluster S by merging with part of it, P, before all of the points in S are joined. If l joins P before any points in $Q = S \cap \overline{P}$, we must have that

$$d_{com}(l, P) < d_{com}(i, P), \qquad \forall i \in Q = S \cap \overline{P}, \tag{7.1}$$

where complete linkage distance d_{com} is given by

$$d_{com}(l, P) \equiv \max_{j \in P} \; d(l, j).$$

Now since $Q \subset S$, and $P \subset S$, we have

$$d_{com}(i, P) \leq \text{diameter}(S) \qquad \forall i \in Q. \tag{7.2}$$

Now (7.1) and (7.2) imply that

$$\exists \;\; l \notin S \text{ such that } \max_{j \in P} d(l, j) < \text{diameter}(S). \tag{7.3}$$

Now

$$\min_{j \in S} d(l, j) < \max_{j \in P} d(l, j),$$

so (7.3) becomes

$$\exists \ \ l \notin S \text{ such that } \min_{j \in S} d(l, j) < \text{diameter}(S). \tag{7.4}$$

But because S is a mutual cluster, we know that

$$\min_{j \in S} d(i, j) > \text{diameter}(S) \qquad \forall i \notin S. \tag{7.5}$$

Thus (7.4) and (7.5) are a contradiction: l cannot be joined to any point in mutual cluster S before any member of S.

This proof applies equally well to single, average, or complete linkage.

References

Chipman, H., George, E. & McCulloch, R. (1998), Making sense of a forest of trees, *in* S. Weisberg, ed., 'Proceedings of the 30th Symposium on the Interface', Interface Foundation of North America, Fairfax Station, VA, pp. 84–92.

Chipman, H., Hastie, T. & Tibshirani, R. (2003), *Clustering microarray data; in Statistical analysis of gene expression microarray data, T.Speed editor*, Chapman and Hall/CRC, pp. 159–199.

Eisen, M., Spellman, P., Brown, P. & Botstein, D. (1998), 'Cluster analysis and display of genome-wide expression patterns', *Proc. Natl. Acad. Sci., USA.* **95**, 14863–14868.

Gersho, A. & Gray, R. (1992), *Vector Quantization and Signal Compression*, Kluwer Academic Publishers, Boston, MA.

Gordon, A. (1999), *Classification (2nd edition)*, Chapman and Hall/CRC press, London.

Hartigan, J. A. & Wong, M. A. (1979), '[(Algorithm AS 136] A k-means clustering algorithm (AS R39: 81v30 p355-356)', *Applied Statistics* **28**, 100–108.

Hastie, T., Tibshirani, R. & Friedman, J. (2001), *The Elements of Statistical Learning; Data mining, Inference and Prediction*, Springer Verlag, New York.

Kaufman, L. & Rousseeuw, P. (1990), *Finding Groups in Data: An Introduction to Cluster Analysis*, Wiley, New York.

Kohonen, T. (1989), *Self-Organization and Associative Memory (3rd edition)*, Springer-Verlag, Berlin.

Lloyd, S. (1957), Least squares quantization in PCM., Technical report, Bell Laboratories. Published in 1982 in IEEE Trans. Inf. Theory **28** 128-137.

Macnaughton-Smith, P., Williams, W., Dale, M. & Mockett, L. (1965), 'Dissimilarity analysis: a new technique of hierarchical subdivision', *Nature* **202**, 1034–1035.

Sorlie, T., Perou, C., Tibshirani, R., Aas, T., Geisler, S., Johnsen, H., Hastie, T., Eisen, M., van de Rijn, M., Jeffrey, S., Thorsen, T., Quist, H., Matese, J., Brown, P., Botstein, D., Lonning, P. & Borresen-Dale, A.-L. (2001), 'Gene expression patterns of breast carcinomas distinguish tumor subclasses with clinical implications', *Proc .Nat. Acad. Sci.* **98**, 10969–74.