Validating Clustering for Gene Expression Data

Ka Yee Yeung
David R. Haynor
Walter L. Ruzzo

Technical Report UW-CSE-00-01-01
January, 2000

Department of Computer Science & Engineering
University of Washington
Seattle, WA 98195

Validating Clustering for Gene Expression Data

Ka Yee Yeung
Department of Computer Science and Engineering
University of Washington
Seattle, WA 98195
kayee@cs.washington.edu

David R. Haynor
Department of Radiology
University of Washington
Seattle, WA 98195
haynor@u.washington.edu

Walter L. Ruzzo
Department of Computer Science and Engineering
University of Washington
Seattle, WA 98195
ruzzo@cs.washington.edu

January 24, 2000

Abstract

Many clustering algorithms have been proposed to analyze gene expression data, but little guidance is available to help choose among them. We provide a systematic and quantitative framework to assess the results of clustering algorithms. A typical gene expression data set contains measurements of the expression levels of a fixed set of genes under various experimental conditions. Clustering algorithms attempt to partition the genes into groups exhibiting similar patterns of variation in expression level, hopefully revealing biologically meaningful patterns of activity or control. Our methodology is to apply a clustering algorithm to the data from all but one experimental condition. The remaining condition is used to assess the predictive power of the resulting clusters—meaningful clusters should exhibit less variation in the remaining condition than clusters formed by coincidence. We have successfully applied the methodology to compare three clustering algorithms on three published gene expression data sets. In particular, we found our quantitative measures of cluster quality to be positively correlated with external standards of cluster quality (functional categorizations of genes known for two of the three data sets).

1 Introduction and Motivation

In an attempt to understand complicated biological systems, large amounts of gene expression data have been generated by researchers (see [3] and [14]). Because of the large number of genes and the complexity of biological networks, clustering is a useful data exploratory technique for gene expression analysis. Many clustering algorithms have been proposed for gene expression data. For example, Eisen *et al.* [5] applied the average linkage hierarchical clustering algorithm to identify groups of co-regulated yeast genes. Ben-Dor *et al.* [1] reported success with their CAST algorithm. Tamayo *et al.* [13] used self-organizing maps to identify clusters in the yeast cell cycle and human hematopoietic differentiation data sets.

Assessing the clustering results and interpreting the clusters found are as important as generating the clusters [7]. In much of the published clustering work on gene expression, the success of clustering algorithms is assessed by visual inspection using biological knowledge (for example, [11] and [5]). The following example (illustrated in Figure 1) shows the importance of assessing clustering results. In Figure 1(a) and Figure 1(b), genes are clustered using the average linkage hierarchical clustering algorithm so that similar genes are placed adjacent to each other along the vertical axis. The experiments or conditions along the horizontal axis are not clustered. The color intensity of each cell in the figure is proportional to the measured gene expression ratio, with bright red representing the most positive and bright green being most negative. Figure 1 (a) shows a cluster identified by Eisen *et al.* (Figure 2E in [5]) from a data set with 2467 yeast genes and 79 conditions. Figure 1 (b) shows a striking pattern obtained in our simulation data set which does *not* contain any intrinsic pattern. The simulation data set, $D(i, j)$, is obtained by randomly choosing a mean expression level α_i for each gene i and randomly choosing a mean value β_j for each condition j according to standard normal distributions. Each entry in the simulation data set, $D(i, j)$, is chosen from the normal distribution with mean $(\alpha_i + \beta_j)$ and variance 1. Therefore, the simulation data set contains no instrinsic pattern. The CLUSTER software [5], which is an average linkage hierarchical clustering algorithm, is then applied to the genes in the simulation data set. Figure 1(b) is a pattern representing a subtree in the dendrogram identified by visual inspection using the TREEVIEW software [5]. Due to some technical difficulties, the resolutions of Figures 1 (a) and (b) are different. Despite the difference in resolution in the two figures, genes in Figure 1(b) show a striking pattern which can easily be interpreted as a potential cluster without any validation procedures. To the best of our knowledge, there is no systematic data-driven method to quantitatively evaluate gene expression clustering results.

Furthermore, the clusters obtained by different clustering algorithms can be remarkably different. A biologist with a gene expression data set is faced with the problem of choosing an appropriate clustering algorithm for his or her data set. Our paper provides a quantitative data-driven framework to evaluate and compare different clustering algorithms.

Many clustering algorithms take a similarity matrix, instead of the raw gene expression data, as input. In the similarity matrix, the similarity between two gene expression series in the raw data is reduced to a single value, called *pairwise similarity*. The pre-clustering step of choosing the similarity metric, with which pairwise similarities are computed, can have a serious impact on clustering results. There are no general guidelines in the literature for the choice of similarity metrics [12]. Our approach can also be used to investigate the effect of similarity metrics on the quality of clustering results.

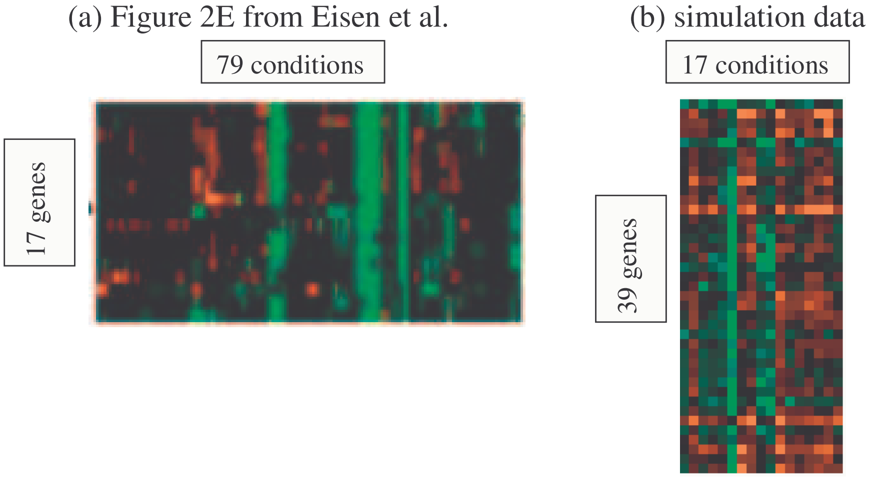


Figure 1: (a) Figure 2E from Eisen *et al.* [5] (b) A striking pattern from our simulation data which contains no intrinsic pattern.

2 Previous Work on Cluster Validation

According to Jain and Dubes [7], *cluster validation* refers to procedures that evaluate the results of cluster analysis in a quantitative and objective fashion. In the statistics literature, cluster validation procedures are divided into two main categories: external and internal criterion analysis [7]. Chapter 4 in Jain and Dubes [7] provides a detailed discussion of the various indices proposed to validate clustering results.

The external criterion analysis validates a clustering result by comparing the clustering result to a given "gold" standard which is another partition of the objects. The "gold" standard can be obtained by an independent process based on information other than the given data set. This criterion will be discussed in more details in Section 6.

The internal criterion analysis uses information from within the given data set to represent the goodness of fit between the input data set and the resulting clustering results.

There are usually some tunable parameters to clustering algorithms which in turn determines the number of clusters produced. Another aspect of cluster validation is to justify the number of clusters in a clustering result. Determining the optimal number of clusters is a very difficult problem [8]. Jain *et al.* [8] used a bootstrapping technique to estimate the optimal number of clusters in a given data set.

3 Problem Statement and Basic Idea

Our work to assess the quality of clustering results is motivated by the *jackknife* approach [4]. A typical gene expression data set contains measurements of expression levels of n genes measured under m experimental conditions. Presumably, the expression levels of co-regulated genes will vary similarly across the m conditions (or experiments), so clustering the genes based on similarities among these expression level measurements should isolate clusters of biologically related genes. Our goal is to compare the clustering results of two (or more) clustering algorithms, say algorithm A and algorithm B. Our idea is to apply a clustering algorithm to the data from $(m-1)$ conditions, and to use the remaining condition to assess the predictive power of the clustering algorithm. A clustering algorithm is said to have good predictive power if genes in the same cluster tend to have similar expression levels in the condition that is not used to produce the clusters. We define a scalar quantity called the *figure of merit* (FOM), which is an estimate of the predictive power of a clustering algorithm.

The idea is illustrated in Figure 2, in which a clustering algorithm is applied to the data from conditions $0, 1, \ldots, (e-1), (e+1), \ldots, (m-1)$, and condition e is used to estimate the predictive power of the algorithm. Suppose k clusters, $C_1, C_2, \ldots, C_k$, are obtained, with cluster sizes $s_1, s_2, \ldots, s_k$, such that $\sum_{i=1}^{k} s_i = n$. Let $R(i, j)$ be the expression level of gene i under condition j in the raw data matrix. Let $FOM(e, k)$ be the figure of merit for k clusters and using condition e as validation. There are many possible definitions of the figure of merit (see Section 5). For example, a possible definition of FOM is the average squared distance from the mean expression level in each cluster, which can be written as $FOM(e, k) = \sqrt{\frac{1}{n} * \sum_{i=1}^{k} \sum_{x \in C_i} (R(x, e) - \mu_{C_i}(e))^2}$, where $\mu_{C_i}(e)$ is the average expression level in condition e of genes in cluster C_i.

Each of the m conditions can be used in turn as the validation condition. The *aggregate figure of merit*, $FOM_{tot}(k) = \sum_{e=0}^{m-1} FOM(e, k)$, is an estimate of the total predictive power of the algorithm over all the conditions for k clusters in a data set.

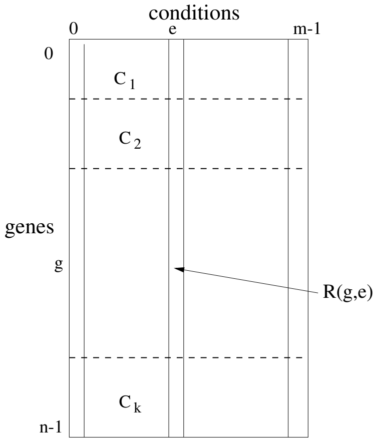


Figure 2: Raw data matrix.

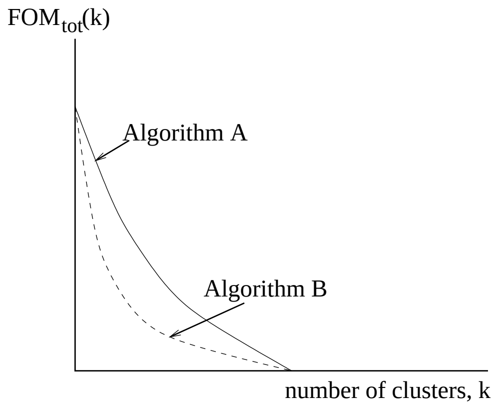


Figure 3: Comparison of algorithms A and B.

In the case of using the average of the squared distance from the mean as the figure of merit, a small aggregate figure of merit is an indication of a high predictive power clustering algorithm. For example, in Figure 3, algorithm B has higher predictive power than algorithm A. Different algorithms typically have different tunable parameters. To compare the quality of clusters produced by two different algorithms, we must adjust the parameters so that the number of clusters is the same in both cases. Otherwise, simply producing more (and therefore smaller) clusters will produce a smaller FOM. Determining the optimal number of clusters is a very difficult problem [8]. Since we cannot determine the optimal number of clusters for a given data set, we cannot produce clustering results with the optimal number of clusters. Therefore, our methodology of comparing clustering algorithms over a range of number of clusters is a reasonable way to get around the problem of determining the optimal number of clusters.

Our approach is different from *leave-one-out cross validation* in machine learning. In leave-one-out cross validation in machine learning, the objective is to estimate the accuracy of a *classifier*, which is an algorithm that maps an unlabelled instance to a label, by *supervised learning* [9]. The labels of the objects to be clustered are assumed to be known. The idea is to hide the label of each object in turn, and to estimate the label of the object using a classifier. This is in contrast to our approach in which we do *not* assume any prior information of the genes to evaluate the quality of clustering results. Instead, we define figures of merit, which are estimators of the predictive power of clustering algorithms, to assess the quality of clustering results.

We demonstrated our technique on three clustering algorithms (CAST [1], k-means and an iterative algorithm) and three gene expression data sets: the rat Central Nervous System (CNS) data set [14], the yeast cell cycle data set [2], and the human hematopoietic differentiation data set [13]. Section 4 describes the clustering algorithms, and Section 5 discusses different definitions for the figure of merit. Section 6 provides a methodology to measure the correspondence of a clustering result with a given partial categorization of genes. We will show that the performance of the clustering algorithms depends on the specific data set, the number of clusters and the definition of figure of merit. None of the three clustering algorithms is a clear winner in our results. In general, the k-means algorithm has comparable average performance to the CAST algorithm. We will also provide evidence that the figure of merit is an estimator for the predictive power of clustering algorithms using external validation with known functional categories of genes. We will also show that the predictive power of the CAST algorithm using the Euclidean distance and the correlation coefficient as similarity metrics are comparable on the rat CNS data set. The detailed results will be described in Section 7. Our main contribution is not the comparison of these specific algorithms and metrics, but rather the development of a simple, quantitative data-driven methodology allowing such comparisons to be made between any clustering algorithms and any similarity metrics.

4 Clustering Algorithms

We implemented three clustering algorithms: the *Cluster Affinity Search Technique* (CAST) [1], an *iterative* partition algorithm[1] and the *K-means* algorithm[7]. For comparison, a random clustering algorithm is also implemented.

4.1 CAST [1]

We implemented the pseudo-code of CAST in [1] with two additional heuristics that have been added to BIOCLUST, the implementation of CAST by its authors. Please refer to [1] for the details of the algorithm. One heuristic is to choose a gene with the maximum number of neighbors to start a new cluster. After the CAST algorithm converges, there is an additional iterative step, in which all clusters are considered at the same time, and genes are moved to the cluster with the highest average similarity.

4.2 Iterative Partition Algorithm

The input to the iterative partition algorithm consists of a similarity matrix S, and a parameter α. Varying the parameter α produces clustering results with different number of clusters. The total similarity of a gene g to a cluster C, $Sim_{tot}(g, C)$, is defined as the sum of the pairwise similarities from g to each gene in C, *i.e.*, $Sim_{tot}(g, C) = \sum_{x \in C} S(g, x)$, where $S(g, x)$ is the pairwise similarity

[1] The iterative algorithm was suggested by Richard M. Karp at University of California, Berkeley.

of gene g and gene x. The *excess similarity* from a gene g to a cluster C, $Excess_{sim}(g, C)$, is defined as the excess of the total similarity from g to C over α multiplied by the size of cluster C, *i.e.*, $Excess_{sim}(g, C) = Sim_{tot}(g, C) - \alpha * |C|$.

Initially, each gene is in its own cluster. A random order is selected for the genes in the iterative step. In each iteration, for each gene g, the excess similarity from gene g to each existing cluster is computed. If C_{max} is the cluster with the maximum excess similarity to gene g and gene g is not currently in cluster C_{max}, gene g is removed from the cluster it is in, and is inserted in cluster C_{max}. This process is repeated until no genes are moved between clusters.

Note that in CAST, there is only one cluster open at a time, while all clusters are open at the same time in the iterative algorithm.

4.3 K-means

The number of clusters, k, is an input to the k-means clustering algorithm. Clusters are described by *centroids*, which are cluster centers, in the algorithm. In our implementation of k-means [7], the initial centroids consist of k randomly chosen genes. Each gene is assigned to the centroid (and hence cluster) with the closest Euclidean distance. New centroids of the k clusters are computed after all genes are assigned. The steps of assigning genes to centroids and computing new centroids are repeated until no genes are moved between clusters.

4.4 Random Clustering

To evaluate the performance of a clustering algorithm, we can compare the clustering algorithm to random clustering. A random clustering for k clusters and condition e can be obtained by randomly putting the data values in condition e into k bins. If the figure of merit obtained from a clustering algorithm is considerably lower than that from random clustering, this is evidence that the clustering algorithm has higher predictive power than random clustering.

5 Figure of Merit

A *figure of merit* is an estimate of the predictive power of a clustering algorithm. Suppose a clustering algorithm A is applied to all conditions except condition e, and there are k clusters. The figure of merit, $FOM(e, k)$, considered in Section 3 is an estimate of the mean error of predicting the expression levels from the average expression levels of the clusters in condition e. Let $R(x, e)$ be the expression level of gene x in condition e, and $\mu_{C_i}(e)$ be the average expression level in condition e of genes in cluster C_i. The *2-norm* figure of merit, FOM_2, is defined as:

$$FOM_2(e, k) = \sqrt{\frac{1}{n} * \sum_{i=1}^{k} \sum_{x \in C_i} (R(x, e) - \mu_{C_i}(e))^2} \tag{1}$$

Similarly, we can define the *1-norm* figure of merit, FOM_1, as the average Manhattan distance between the mean expression level in each cluster and the expression levels of genes in the cluster in condition e.

$$FOM_1(e, k) = \frac{1}{n} * \sum_{i=1}^{k} \sum_{x \in C_i} |R(x, e) - \mu_{C_i}(e)| \tag{2}$$

Define the *range* in condition e of cluster C_i as the difference of the maximum expression level, $max_{C_i}(e)$, and the minimum expression level, $min_{C_i}(e)$, in condition e of cluster C_i. The range measures the diameter of a cluster. The average range of the k clusters is defined as the *range* FOM, FOM_R.

$$FOM_R(e,k) = \frac{1}{k} * \sum_{i=1}^{k} (max_{C_i}(e) - min_{C_i}(e)) \tag{3}$$

The *minimum achievable figure of merit* for condition e is the minimum possible FOM given the data values in condition e only. The minimum achievable figure of merit represents the lower bound of the figure of merit of any clustering algorithm.

Theorem 1: The minimum achievable figure of merit for the range FOM, $FOM_R^{min}(e,k)$, can be computed in $O(n \log n)$ time.
Proof Outline:
First, we proved by contradiction that the optimal solution must be a partition of sorted data values. In the second part of the proof, we showed that the cluster boundaries should be placed in the $(k-1)$ largest gaps in the sorted data values in condition e since $FOM_R^{min}(e,k)$ is minimized when the gap values between sorted data values are maximized. Therefore, $FOM_R^{min}(e,k)$ can be computed by sorting the data values in condition e, and placing the cluster boundaries in the $(k-1)$ largest gaps between sorted data values. This can be done in $O(n \log n)$ time. $\square$

The 2-norm, 1-norm and range FOM estimate the predictive power of a clustering algorithm by measuring the dispersion of the gene expression levels in the left-out condition e. Intuitively, genes in the same clusters are expected to have similar expression levels. Moreover, disjoint clusters are expected to be relatively far apart from each other. Therefore, we can define the *ratio FOM* to be the ratio of the within-cluster dispersion to the between-cluster separation. The within-cluster dispersion can be represented by the 1-norm FOM. The between-cluster separation can be represented by the distance between the centers of the k clusters in condition e. Let $\mu_{C_i}(e)$ be the average gene expression level of cluster C_i in condition e. Denote the maximum average gene expression level in condition e by $\mu_{C_i}^{max}(e) = \max_{C_i} \mu_{C_i}(e)$. Similarly, the minimum average gene expression level in condition e is $\mu_{C_i}^{min}(e) = \min_{C_i} \mu_{C_i}(e)$. The average between-cluster separation can be estimated by the average Manhattan distance between cluster means, *i.e.*, $\frac{\mu_{C_i}^{max}(e) - \mu_{C_i}^{min}(e)}{k-1}$. Hence, the *ratio FOM* can be written as:

$$FOM_{ratio}(e,k) = \frac{\frac{1}{n} * \sum_{i=1}^{k} \sum_{x \in C_i} |R(x,e) - \mu_{C_i}(e)|}{\frac{1}{k-1} * (\mu_{C_i}^{max}(e) - \mu_{C_i}^{min}(e))} \tag{4}$$

6 External Validation of FOM

In this section, we will describe a methodology to justify the use of figures of merit as a measure of the predictive power of clustering algorithms. Suppose the functional categories of a subset of the genes in the given data set are known. Although the functional categorization may be derived from information other than gene expression data, a gene expression data set is expected to reflect the functional categories to a certain degree. The idea is to compare a clustering result to a given known functional categorization of the genes. Since not all genes have been classified, we allow a *partial* functional categorization as the "gold" standard to compare clustering results with. This methodology can also be applied to determine the similarity of two clustering results.

6.1 The Jaccard and Hurbert indices

In Chapter 4 of Jain and Dubes [7], *external indices* that assess the degree to which two partitions with the same number of objects agree are defined. Since clustering algorithms assign each object to exactly one cluster, clustering results can be considered as partitions. Suppose two partitions, U and V, are to be compared. Jain and Dubes [7] define a to be the number of pairs of objects that appear in the same cluster in both partitions, b to be the number of pairs of objects in the same cluster in U but not in V, c to be the number of pairs of objects in the same cluster in V but not in U, and d to be the number of pairs of objects in different clusters in both partitions. There are several known indices that measure the similarity of two partitions, for example, the Jaccard index:

$$Jaccard = \frac{a}{a+b+c} \tag{5}$$

They also define $m_1 = a + b$, which is the number of pairs of objects that are in the same group in U. Similarly, $m_2 = a + c$ is the number of pairs of objects in the same group in V. Let $M = a + b + c + d$. The Hurbert Γ statistic is defined as:

$$Hurbert = \frac{Ma - m_1 m_2}{\sqrt{m_1 m_2 (M - m_1)(M - m_2)}} \tag{6}$$

The Hurbert Γ statistic is essentially the correlation between two matrices I_U and I_V. $I_U(i, j) = 1$ if object i and object j are in the same group in partition U, and $I_U(i, j) = 0$ otherwise. I_V is similarly defined.

The Jaccard index lies between 0 and 1, while the Hurbert Γ statistic lies between -1 and 1. A high Jaccard or Hurbert Γ statistic means a high correspondence to the given functional categorization.

6.2 Our Generalization

In Jain and Dubes [7], they assume the partitions to be compared are *complete* partitions of all the objects. We generalize the indices in Section 6.1 to compare *partial* partitions since not all genes fall into known functional categories. The main observation is that the Jaccard and Hurbert Γ statistics only depend on the values a, b, c, d. Our idea is that we only count the number of pairs of genes such that both genes exist in the given known categorization. Suppose U is a clustering result of n genes, and V is a *partial* functional categorization of r genes, where $r \leq n$. We define a to be the number of pairs of genes such that both genes exist in V, and both genes appear in the same group of V and same cluster in U. We can similarly define b, c and d.

A high Jaccard or Hurbert Γ statistic indicates high similarity of a clustering result to a given functional categorization. Recall that a low figure of merit indicates high predictive power. The idea is to apply a clustering algorithm to all conditions except condition e to produce k clusters, $FOM(e, k)$ and the Jaccard or Hurbert Γ statistic of the clustering result based on all conditions except e are computed. Repeat this process for different clustering algorithm, and then plot the statistics against $FOM(e, k)$'s for different algorithms. If the points show a trend of downward negative slope, this shows that a clustering result with a low FOM tend to have a high correspondence to the given functional categorization. This provides evidence for the predictive power of FOM for comparing clustering algorithms.

7 Results and Discussion

In this section, we will describe the performance of the iterative, CAST, and k-means clustering algorithms on the rat CNS data set [14], the yeast cell cycle data set [2], and the human hematopoietic differentiation data set [13]. We will also justify the use of figures of merit as an estimate of the predictive power of clustering algorithms using known biological classes of genes in the rat CNS data set and the yeast cell cycle data set. The three gene expression data sets are available via the World Wide Web. In our experiments, the random clustering step was repeated 1000 times, the iterative algorithm were run 10 times and the k-means algorithm were run 30 times to obtain reliable $FOM(e,k)$'s.

7.1 The Rat CNS Data Set

Figures 4, 5, 6, 7 show the performance of the iterative, k-means and the CAST clustering algorithms in terms of the aggregate 2-norm, 1-norm, range and ratio FOM's on the rat CNS data set. The raw data set published by Wen *et al.* consists of 112 genes and 9 time points. As suggested in [14], the raw data is normalized by the maximum expression level for each gene. The data set is then augmented with slopes (differences between consecutive time points) to capture parallel trajectories of the time course data. This results in a data set with 112 genes and 17 conditions. The correlation coefficient is used to compute pairwise similarities of genes. All three clustering algorithms (iterative, k-means and CAST) achieve lower aggregate figures of merit than random clustering in the 2-norm, 1-norm, range and ratio FOM's. Since the iterative, the k-means and the random clustering algorithms are randomized, each of them is run multiple times to obtain reliable $FOM(e,k)$'s. In the following figures, the solid lines for the iterative, k-means and random algorithms represent the sum of the average $FOM(e,k)$ over all the conditions e. We also show the 80% and 20% error bars in Figure 4. The 80% error bars for the randomized algorithms are obtained by computing the 80 percentile of the $FOM(e,k)$ from the multiple runs of the algorithms, and then summing over all the conditions to obtain the aggregate FOM. Similarly, the 20% error bars are obtained by computing the 20 percentile from the multiple runs. We have chosen to present the 20% and 80% error bars instead of the maximum and minimum because the maximum and minimum are very sensitive to outliers. From Figure 4, we can see that the spread of the FOM of the iterative algorithm is much smaller than that of the k-means and the random clustering algorithms. The error bars of the 1-norm and range FOM show similar behavior and are not be presented here.

The behavior of the clustering algorithms on the aggregate 1-norm FOM (Figure 5) is very similar to that of the 2-norm (Figure 4). When the number of clusters is small (below 25), the CAST and k-means clustering algorithms have comparable aggregate 1-norm and 2-norm FOM's, which are lower than those of the iterative algorithm. When the number of clusters is large (above 30), all three clustering algorithms have comparable aggregate 2-norm and 1-norm FOM's.

In Figure 6, the minimum FOM corresponds to the aggregate minimum achievable range FOM. $FOM_R^{min}(e,k)$ can be computed with Theorem 1. The iterative algorithm has lower aggregate range FOM's than the CAST and k-means algorithms when the number of clusters is small (below 20). When the number of clusters is large (above 20), all three clustering algorithms have similar aggregate range FOM's. Moreover, all three algorithms produce aggregate range FOM's lying roughly halfway between that of random clustering and the aggregate minimum achievable range FOM.

The 2-norm, 1-norm and range FOM's are expected to be monotonically decreasing as the number of clusters increases. This is not the case for the ratio FOM since it measures the ratio of the within-cluster dispersion to between-cluster dispersion. Small ratio FOM's are desirable. However, the ratio FOM may not be monotonic. Jain *et al.* [8] applied the bootstrap technique to determine the optimal number of clusters. They defined a similar ratio of within-cluster dispersion to between-cluster separation, plotted the ratio against the number of clusters, and argued that a "significant" knee in the graph corresponds to the optimal number of clusters. As a matter of fact, plotting an evaluation index against the number of clusters is a standard technique [7]. In Figure 7, there is a knee around four to six clusters for all of the iterative, k-means and CAST clustering algorithms. The ratio of within-cluster dispersion to between-cluster separation is a minimum around four to six clusters in the rat CNS data set. In [14], the genes in the rat CNS data set are classified into four categories using biological knowledge. The ratio FOM may give us an estimate of the optimal number of clusters inherent in the data.

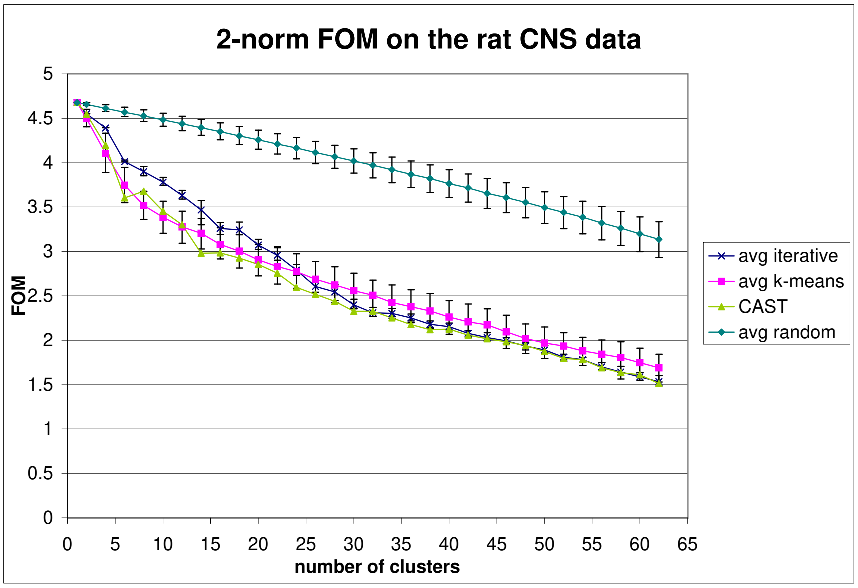


Figure 4: Aggregate 2-norm FOM's of clustering algorithms on the rat CNS data set.

Wen *et al.* [14] categorized genes in the rat CNS data set into four families using biological knowledge. Table 1 shows that the aggregate 2-norm and 1-norm FOM's of the four clusters identified in [14] are comparable to those of the iterative, k-means and CAST algorithms. The k-means algorithm achieves the lowest aggregate 2-norm and 1-norm FOM's for four clusters in our results. It is interesting to observe that the aggregate range FOM from the iterative algorithm is significantly lower than that from the clusters identified by Wen *et al.*. In fact, the aggregate range FOM from Wen's clusters are only slightly lower than that from random clustering. Since our definition of the range FOM depends only on the maximum and minimum expression levels in each cluster, the range FOM is expected to be very sensitive to outliers.

Wen *et al.* [14] found six clusters on this data set with the FITCH software [6]. The CAST algorithm achieves the lowest aggregate 2-norm and 1-norm FOM's for six clusters in our results. The aggregate 2-norm and 1-norm FOM's (data not shown here) of the six clusters found in [14] are slightly lower than those from the CAST algorithm.

Overall, we take the fact that the figures of merit for the clusters chosen by Wen *et al.* are similar to those found algorithmically to be a good indication that our methodology is providing a meaningful

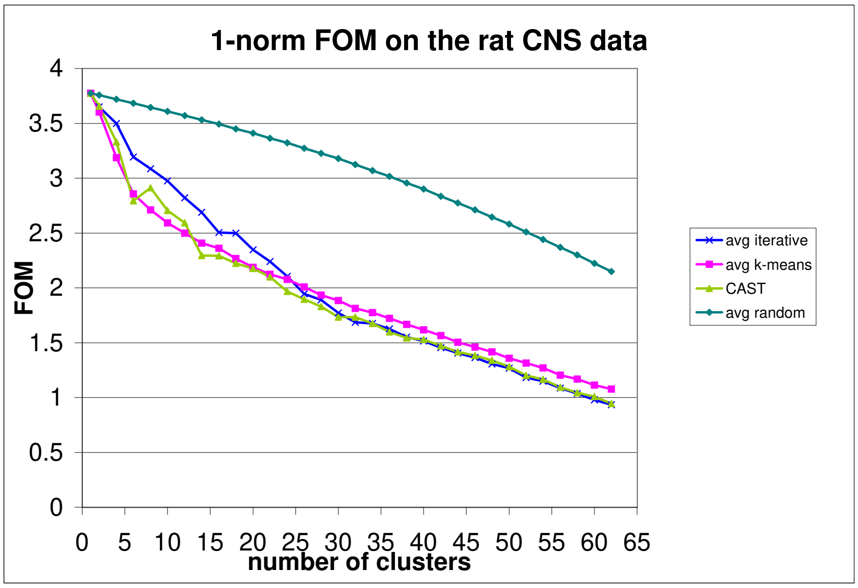


Figure 5: Aggregate 1-norm FOM's of clustering algorithms on the rat CNS data set.

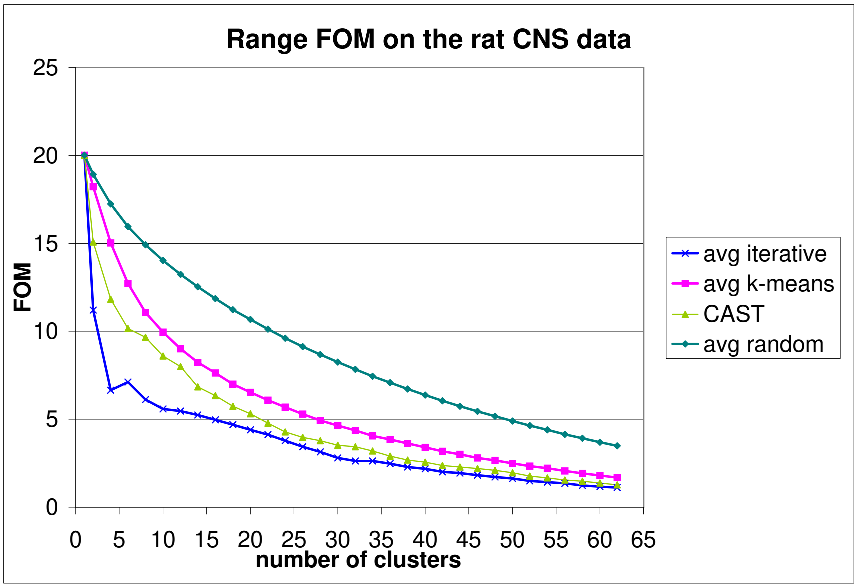


Figure 6: Aggregate range FOM's of clustering algorithms on the rat CNS data set.

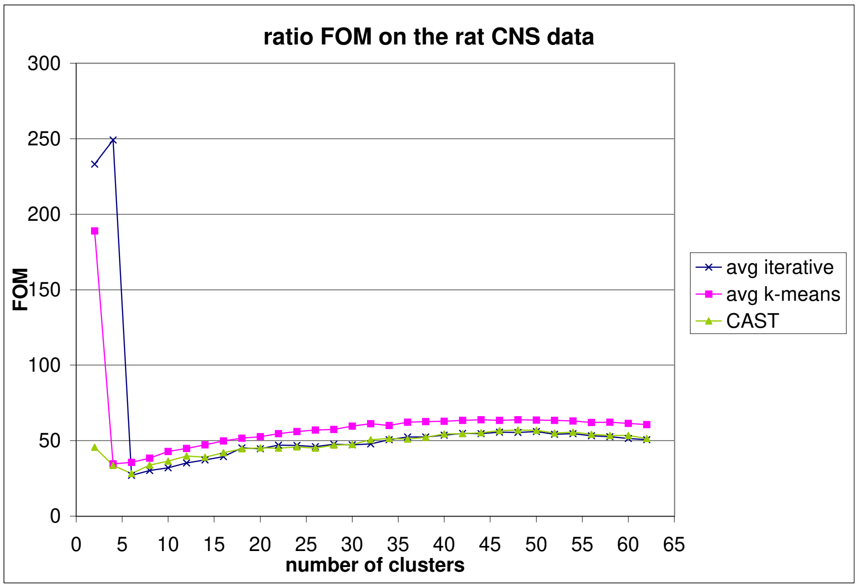


Figure 7: Aggregate ratio FOM's of clustering algorithms on the rat CNS data set.

FOM definition	*Wen's clustering*	*iterative*	*k-means*	*CAST*	*random*
2-norm	4.39	4.43	4.12	4.27	4.61
1-norm	3.43	3.54	3.14	3.40	3.72
range	16.93	5.64	14.97	12.58	17.24

Table 1: Aggregate FOM's from Wen's clusters and from clustering algorithms.

estimate of cluster quality.

In addition to the above FOM calculation, we also evaluated the similarity of our clustering results to the four functional categories of the genes on the rat CNS data set according to Wen *et al.* [14] using the methodology in Section 6.

Figure 8 and Figure 9 show the Jaccard index against $FOM_2(0,4)$ and the Hurbert Γ statistic against $FOM_2(0,4)$ respectively. The iterative, k-means, CAST and random algorithms are applied to 16 conditions (the first condition is left out) of the 112 genes in the rat CNS data set to produce four clusters. The FOM is computed using the gene expression levels in the left-out condition. The iterative and random algorithms are run 10 and 30 times respectively. The CAST algorithm is run exactly once since it is deterministic. The k-means algorithm is run 10 times, and we show the intermediate results of k-means and CAST in successive iterations in Figure 8, Figure 9 and Figure 10.

In Figure 8, Figure 9 and Figure 10, the FOM's of the random algorithm are a few standard deviations higher than the other clustering algorithms, and its Jaccard or Hurbert Γ statistics are a few standard deviations lower than the other clustering algorithms. Moreover, there is an obvious downward negative slope trend in all three figures, showing that clustering results with low FOM's tend to have high correspondence with the given functional categorization.

Figure 10 shows the results of five runs of k-means and one run of the CAST algorithm. Each run of k-means is represented by the same color. The points with the same color represent successive iterations of the same run. Successive iterations of an algorithm show how the FOM's and the

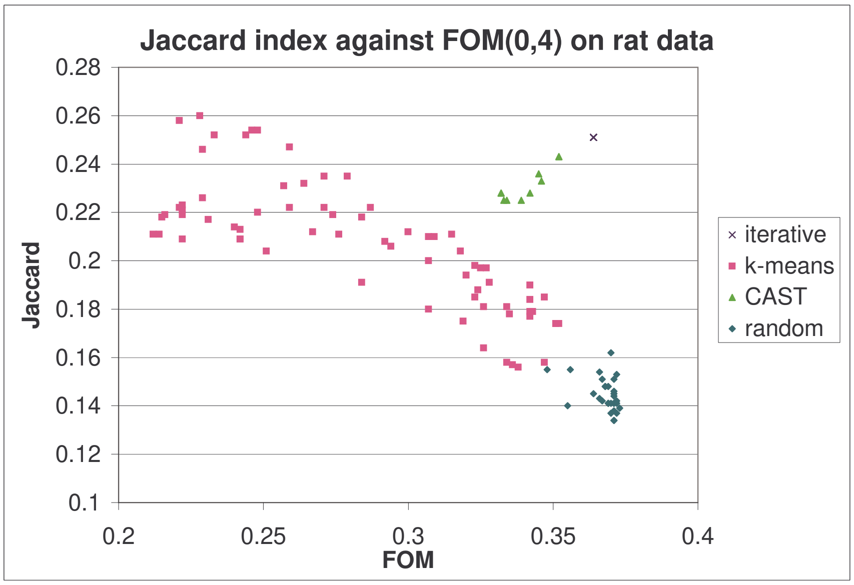


Figure 8: The Jaccard index against $FOM_2(0, 4)$ on the rat CNS data set.

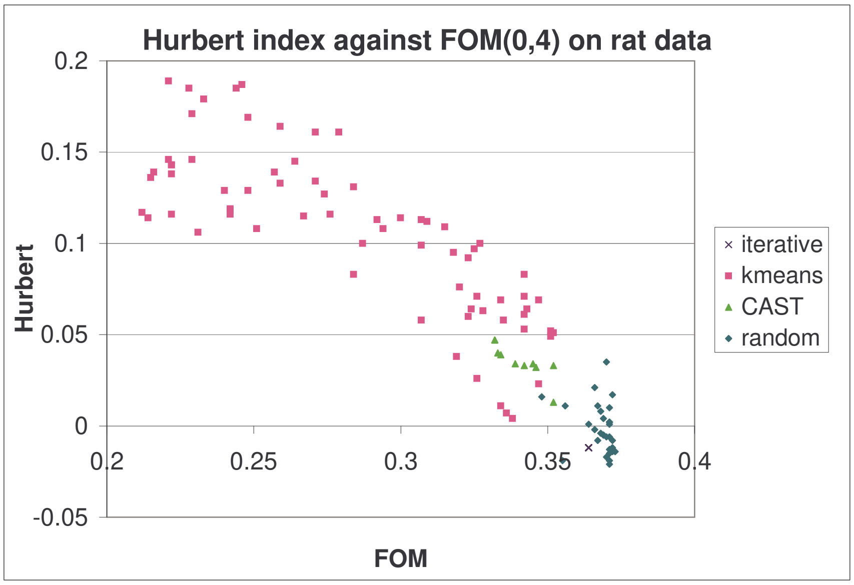


Figure 9: The Hurbert index against $FOM_2(0, 4)$ on the rat CNS data set.

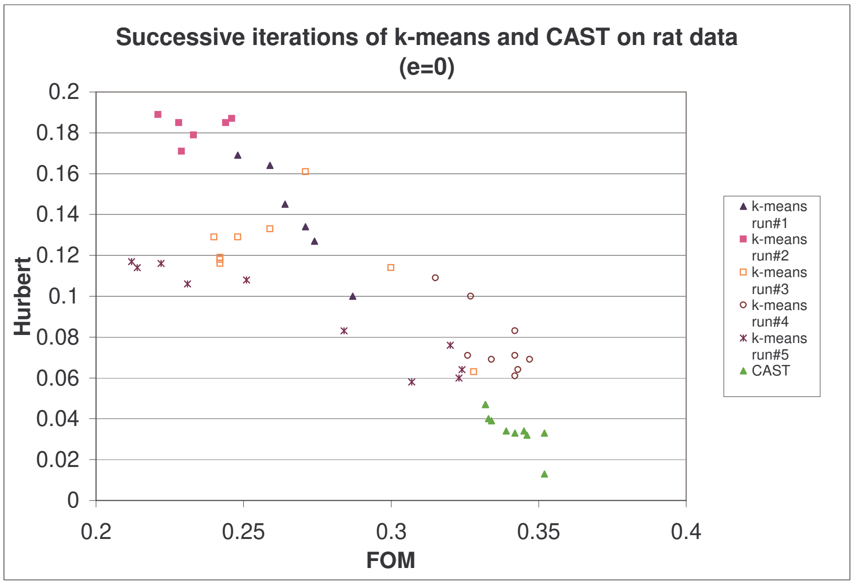


Figure 10: The Hurbert index against $FOM_2(0,4)$ of successive iterations of k-means and CAST on the rat CNS data set.

statistics change as the algorithm proceed to a more desirable clustering result. For most runs of the k-means and CAST, as the algorithm proceed to the next iteration, we can see a trend of lower FOM and a higher correspondence to the given functional categorization.

We also investigated the effect of leaving out other conditions, and found that the shown figures for leaving out the first condition ($e = 0$) are typical results. Furthermore, we computed the average Jaccard and Hurbert Γ statistics when all 17 conditions in the data set are used in clustering algorithms (shown in Table 3) over multiple runs of the algorithms. The iterative and random algorithms are run 10 times, while the k-means algorithm is run 30 times for the results in Table 3. The Jaccard and Hurbert Γ statistics when all 17 conditions are applied are comparable to those when one condition is left out. This shows that leaving out one condition does not have any significant effect on the quality of clustering results.

algorithm	*Jaccard*	*Hurbert*
iterative	0.25	-0.012
k-means	0.24	0.17
CAST	0.28	0.18
random	0.15	-0.001

Table 3: Average Jaccard and Hurbert Γ statistics for all 17 conditions and four clusters on the rat CNS data set.

Hence, we conclude that clustering results with low FOM's tend to have high correspondence to the functional categorization in Wen *et al.* on the rat CNS data set.

7.2 The Yeast Cell Cycle Data Set

The yeast cell cycle data set of Cho *et al.* [2] consists of approximately 6000 genes and 17 conditions. Cho *et al.* [2] identified 420 genes by visual inspection of the raw data. The data set is normalized as in [13]. The 17 conditions are divided into two panels (which correspond to two cell cycles) and are normalized to have mean 0 and variance 1 within each panel. The correlation coefficient is used to compute the similarity matrix.

Figure 11 shows the performance of the clustering algorithms on the processed yeast cell cycle data set. When the number of clusters is large (above 50), the iterative and CAST algorithms have lower aggregate 2-norm FOM's than the k-means algorithm.

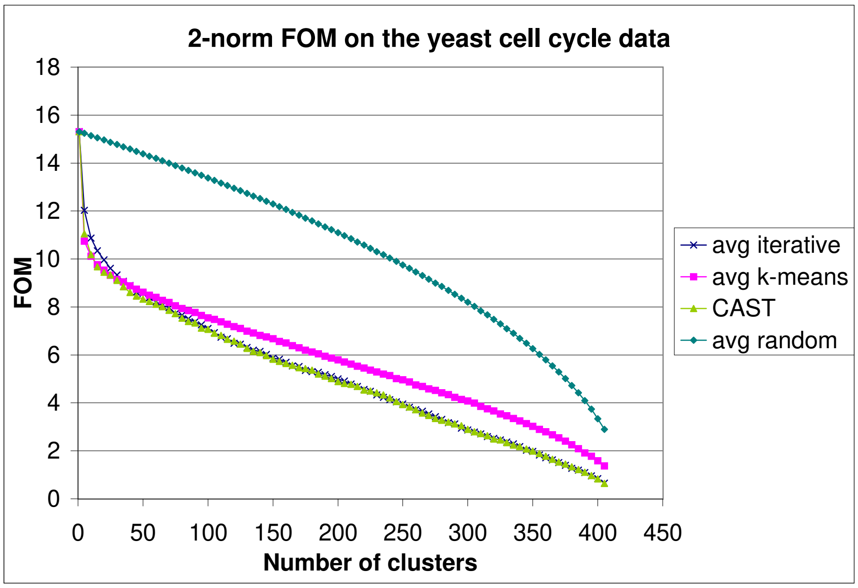


Figure 11: Aggregate 2-norm FOM's of clustering algorithms on the yeast cell cycle data set.

Cho *et al.* categorized approximately 380 genes into five phases of cell cycle on their web site. Since the 420 genes are identified by visual inspection of gene expression data according to the peak times of genes, we expect clustering results to correspond to the five known categories of genes. The methodology of Section 6 is used to validate the use of FOM as an estimate of the predictive power of clustering algorithms. The results for leaving out the first time point ($e = 0$) are shown in Figure 12, Figure 13 and Figure 14. We also studied the effects of leaving out other time points in the data set and found that the shown figures are typical results (not shown here). The randomized algorithms are run 10 times, and the intermediate results of successive iterations of k-means and CAST are also shown.

There is an obvious downward negative slope Figure 12, Figure 13 and Figure 14. The iterative, k-means and CAST algorithms show significantly lower FOM's and higher Jaccard and Hurbert Γ statistics than the random algorithm. Figure 14 shows the FOM's and Hurbert Γ statistics of successive iterations of the k-means and CAST algorithms. Successive iterations of the same run are represented by the same color. In each iteration of k-means and CAST, the FOM's tend to be lower and have higher correspondence to the five functional categories.

We have also computed the average Jaccard and Hurbert Γ statistics when all 17 time points are

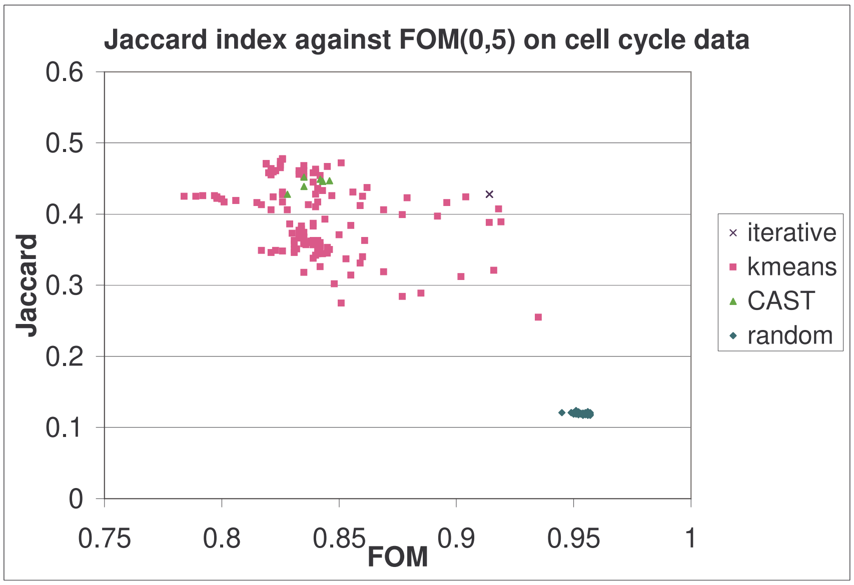


Figure 12: The Jaccard index against $FOM_2(0, 5)$ on the yeast cell cycle data set.

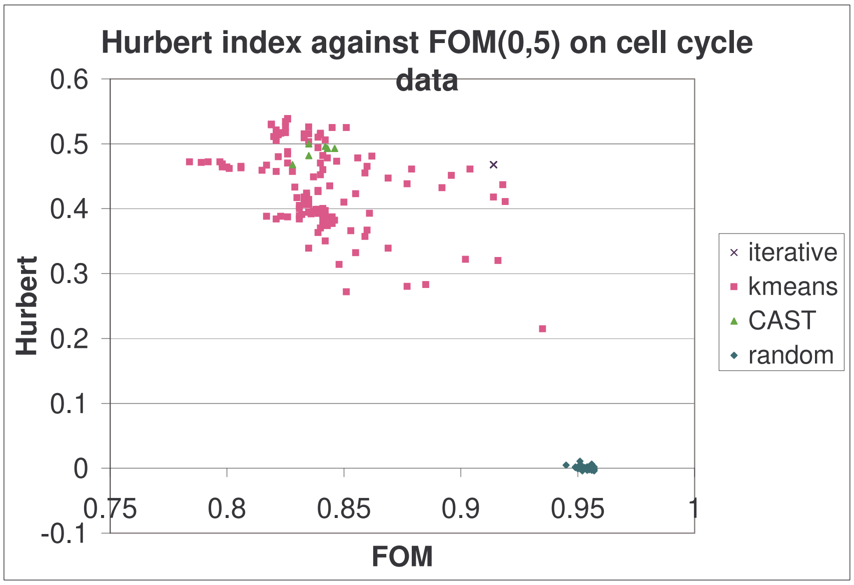


Figure 13: The Hurbert index against $FOM_2(0, 5)$ on the yeast cell cycle data set.

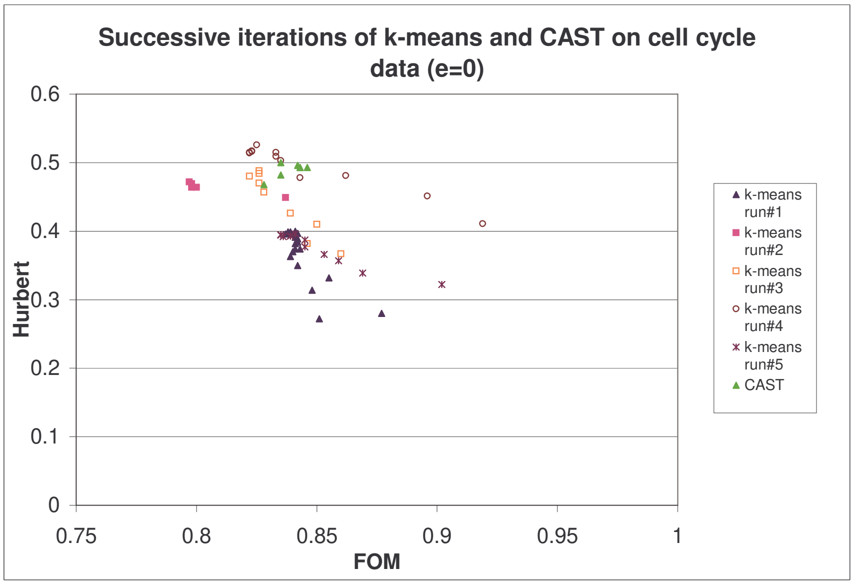


Figure 14: The Hurbert index against $FOM_2(0, 5)$ of successive iterations of k-means on the yeast cell cycle data set.

used in clustering algorithms over multiple runs of the algorithms as shown in Table 4. The average Jaccard and Hurbert Γ statistics when all 17 time points are used are comparable to those when one time point is left out. This is evidence that leaving out one condition does not seriously affect clustering quality on this data set.

algorithm	*Jaccard*	*Hurbert*
iterative	0.42	0.45
k-means	0.43	0.48
CAST	0.45	0.50
random	0.12	0.001

Table 4: Average Jaccard and Hurbert Γ statistics for all 17 conditions and five clusters on the yeast cell cycle data set.

Cho *et al.* [2] also looked up functional categorizations of the 420 genes from the MIPS [10] database. Approximately 180 genes are divided into nine categories. Since the 420 genes are not chosen due to the functional categories in MIPS, and the MIPS database is annotated based on more than just gene expression data, we expect clustering results on the gene expression data to have lower Jaccard and Hurbert Γ statistics with the MIPS functional categories than with the five stages of cell cycle. A typical result is shown in Figure 15. As expected the statistics have lower values than in Figure 13. Note that there is also an obvious downward negative slope in the graph, showing that low FOM's correspond to high correspondence to the MIPS functional categories.

7.3 The Human Hematopoietic Data Set

The aggregate 2-norm FOM's of the clustering algorithms on the human hematopoietic differentiation data set [13] are shown in Figure 16. The variation filter in the GENECLUSTER software [13] is

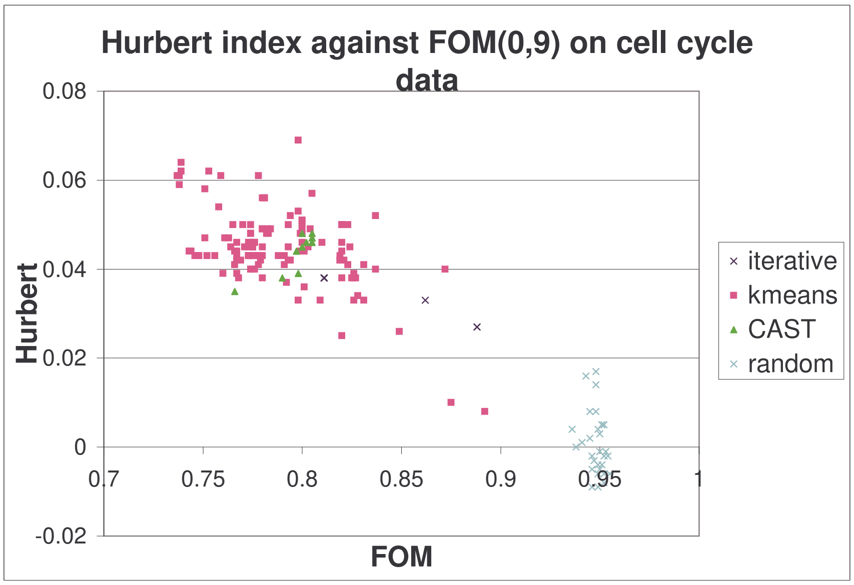


Figure 15: The Hurbert index against $FOM_2(0,9)$ on the yeast cell cycle data set.

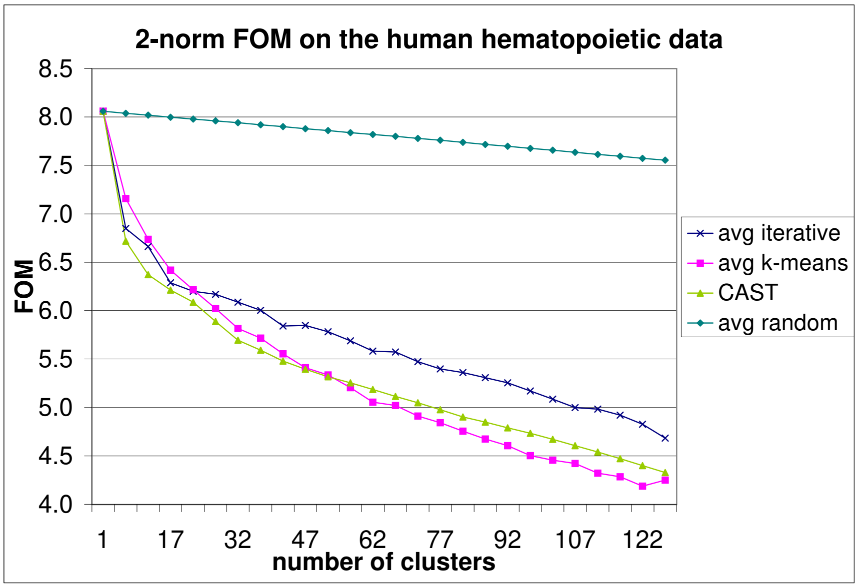


Figure 16: Aggregate 2-norm FOM's of clustering algorithms on the human hematopoietic differentiation data set.

applied to the raw data, and 1033 genes pass through the filter. The data is then normalized across each of the four cell lines making up the 17 conditions as suggested in [13]. The correlation coefficient is used to compute the similarity matrix. A close inspection shows that CAST achieves lower aggregate 2-norm FOM's than the other two algorithms when the number of clusters is small (below 30). When the number of clusters is large (above 50), the k-means algorithm achieves lower aggregate 2-norm FOM's than the iterative and CAST algorithms.

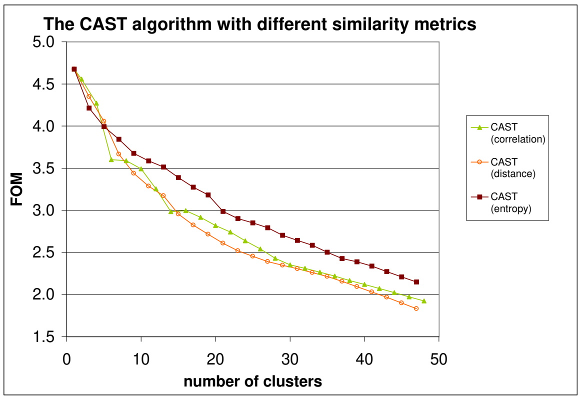
7.4 Analysis

With isolated exceptions, all the data sets we have considered exhibit declining figures of merit under all algorithms as the number of clusters increases. Two factors contribute to this. First, the algorithms may be finding higher quality clusterings, as they subdivide large, coarse clusters into smaller, more homogeneous ones. Second, simply increasing the number of clusters will tend to decrease the FOM. The following simple analysis estimates the effect of the second factor. Suppose the measured expression levels $x_1, \ldots, x_s$ of the s genes in some cluster C are independent, identically distributed normal random variables with variance σ^2. Let $\bar{x} = \sum_{i=1}^{s} x_i/s$. Then C's expected contribution to the FOM_2 is the expected value of $\sum_{i=1}^{s}(x_i - \bar{x})^2$, which is $(s-1)\sigma^2$. Subdividing C into k smaller nonempty sub-clusters would reduce these genes' expected contribution to the FOM_2 to $(s-k)\sigma^2$, and hence the aggregate 2-norm FOM for a collection of k such homogeneous clusters would be $m\sqrt{(n-k)/n}\sigma$. In fact, on the rat CNS data set, this formula agrees with the measured 2-norm $FOM_{tot}(k)$ of random clustering to within a fraction of one percent (despite the fact that the real data sets violate key assumptions in the analysis). Presumably, then, the relatively steep decline in the 2-norm $FOM_{tot}(k)$ achieved by all three clustering algorithms on the rat CNS data set for k up to 20 or so reflect genuine progress in producing more homogeneous clusters, whereas the more gradual declines for larger k, roughly paralleling the random curves, largely reflect the purely statistical effect of increasing k. Although Wen *et al.* only identified 4 to 6 clusters in the rat CNS data set, this analysis suggests that the data may support a more refined sub-clustering.

7.5 Effect of Similarity Metrics

Our approach can also be used to evaluate the effect of similarity metrics on clustering results. There are no general guidelines in the literature for the choice of similarity metrics [12]. Figure 8 shows the aggregate 2-norm FOM's of the CAST and iterative algorithms on the rat CNS data set using the correlation coefficient, Euclidean distance and information entropy as similarity metrics.

Michaels *et al.* [11] proposed to use information entropy as a similarity metric, and they compared their biological knowledge to the clustering results using information entropy and Euclidean distance as similarity metrics with the FITCH software [6]. In their paper [11], they showed that clustering results using Euclidean distance and information entropy have a high degree of correspondence. In order to compute the information entropy of each gene expression sequence, the expression levels are discretized into three equidistant bins. Let I and J be two gene expression series. The information entropy, $H(I), H(J)$, and mutual information, $M(I,J)$, can be computed from the probabilities, $P(i)$, of the occurrence of one of the three discretized expression levels: $H(I) = -\sum[P(i) * \log P(i)]$, and $M(I,J) = H(I) + H(J) - H(I,J)$. The normalized mutual information, M_{norm}, is defined as $M_{norm}(I,J) = M(I,J)/\max\{H(I), H(J)\}$. M_{norm} is a measure of pairwise similarity between two gene expression series.



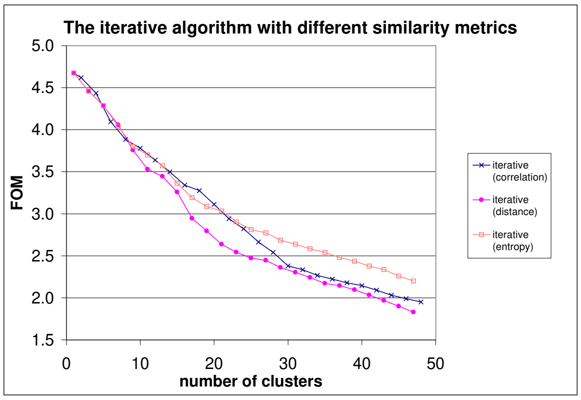


Figure 8: Aggregate 2-norm FOM's of the CAST and iterative algorithms using different similarity metrics.

In Figure 8, when the number of clusters is large (above 30), the aggregate 2-norm FOM's of both algorithms using the correlation coefficient and the Euclidean distance are very similar, and are lower than those of the information entropy. None of the similarity metrics give clearly superior FOM's, especially when the number of clusters is small. The CAST algorithm with the correlation coefficient gives the lowest aggregate 2-norm FOM's for five to eight clusters.

In other experiments (data not shown here), we observed that there is no significant difference in the aggregate 2-norm FOM's of the iterative and CAST algorithms when ten bins are used in the discretization of the expression levels for the information entropy computation in the rat CNS data set.

8 Conclusions

In this paper, we provide a simple and quantitative methodology to compare the predictive power of any clustering algorithms and similarity metrics on any data set. We demonstrated our technique using the iterative, k-means and CAST [1] algorithms on the rat CNS data set [14], the yeast cell cycle data set [2], and the human hematopoietic differentiation data set [13]. We showed that the performance of clustering algorithms depends on the specific data set, the number of clusters, and the definition of FOM. On the rat CNS data set and the yeast cell cycle data set, we showed that low $FOM(e, k)$'s tend to correspond to a high similarities to known partial functional categorizations of genes. This is a good indication that our definitions of figures of merit provide a good estimate of cluster quality. We found that the 1-norm and 2-norm FOM's have very similar performance on all three data sets. Since the range FOM is very sensitive to outliers and the aggregate range FOM's from the iterative algorithm tend to be significantly lower than the other algorithms (especially when the number of clusters is small), this suggests that the iterative algorithm may be more suited to handle data with a lot of outliers.

Our methodology can also be used to verify the existence of clusters in a given data set. If we plot the FOM against the number of clusters for the simulation data set in Section 1, the diagram shows a different trend than data sets with intrinsic patterns (diagram not shown here). For the simulation data with no intrinsic pattern, the trend of all clustering algorithms follow more closely to the random clustering algorithm.

No clustering algorithm emerged as a clear-cut winner in this work, and we suggest that flexibility, speed, reliability and ease of implementation may be equally important in differentiating clustering algorithms. In our implementation, k-means is substantially faster than the iterative and CAST algorithms. It takes under two seconds on a Pentium II 400 to run k-means once on the human hematopoietic data set (1033 genes and 17 conditions). The average running times of the iterative and CAST algorithms are over 50 seconds on the same data set. In terms of reliability, CAST is implemented as a deterministic algorithm, but the iterative and k-means algorithms are randomized algorithms. The error bars in Figure 4 showed that the standard deviations of the iterative algorithm tend to be much smaller than those of the k-means algorithm.

More work needs to be done to confirm the apparent small effect of the predictive power on the similarity metric used to measure similarity of gene expression levels. There are many directions of future work, one of which is to compare the similarity of clustering results of different algorithms. For example, given two genes x and y that are in the same cluster obtained by algorithm A, it would be interesting to compute the probability that x and y are in the same cluster if algorithm B is applied. Another important pre-clustering step is the standardization of variables in the data set. The effect of different variable standardization methods on the predictive power of clustering algorithms would be another interesting direction of future work.

To summarize, clustering is a difficult problem. It would be nice if there were a single universally superior clustering method. However, given the observed variability in the solutions produced by the different algorithms on different data sets and under varying similarity metrics, no such solution is in sight. Lacking that, we feel that the simple methodology introduced in this paper for quantitative comparison of the predictive power of clustering algorithms will prove to be a valuable ingredient in future clustering studies.

Acknowledgement

We would like to thank Richard M. Karp for suggesting the iterative algorithm. We would also like to thank Amir Ben-Dor for sharing the additional heuristics implemented in their software with us. In addition, we would like to thank Lue Ping Zhao at the Fred Hutchinson Cancer Research Centre for his suggestions on modelling simulation data sets. Many colleagues and friends at University of Washington provided valuable suggestions to us: Corey Anderson, Mathieu Blanchette, Jeremy Buhler, Jared Saia, Andrew Siegel, Martin Tompa, and the array analysis group. We would like to thank the Whitehead Institute for granting us an academic site license for their GENECLUSTER software. We would also like to thank Michael Eisen and the Stanford University for granting us the permission to use their CLUSTER and TREEVIEW softwares.

References

[1] A. BEN-DOR, Z. YAKHINI. *Clustering gene expression patterns.* Proceedings of the Third International Conference on Computational Biology (Recomb 99), pp. 33-42, 1999.

[2] R. J. CHO, M. J. CAMPBELL, E. A. WINZELER, L. STEINMETZ, A. CONWAY, L. WODICKA, T. G. WOLFSBERG, A. E. GABRIELIAN, D. LANDSMAN, D. J. LOCKHART, R. W. DAVIS. *A genome-wide transcriptional analysis of the mitotic cell cycle.* Molecular Cell, vol 2, pp. 65-73, July 1998.

[3] J. L. DERISI, V. R. IYER, P. O. BROWN. *Exploring the metabolic and genetic control of gene expression on a genomic scale.* Science, vol. 278, pp. 680-686, 1997.

[4] B. EFRON. *The jackknife, the bootstrap, and other resampling plans.* Society for Industrial and Applied Mathematics, 1982.

[5] M. B. EISEN, P. T. SPELLMAN, P. O. BROWN, D. BOTSTEIN. *Cluster analysis and display of genome-wide expression patterns.* PNAS, vol 95, pp. 14863-14868, Dec 1998.

[6] J. FELSENSTEIN. *PHYLIP (Phylogeny Inference Package), version 3.5c*, distributed by the author, Department of Genetics, University of Washington, Seattle. (1993)

[7] A. K. JAIN, R. C. DUBES. *Algorithms for clustering data.* Prentice Hall, Englewood Cliffs, NJ, 1988.

[8] A. K. JAIN, J. V. MOREAU. *Bootstrap technique in cluster analysis.* Pattern Recognition, vol 20, no. 5, pp. 547-568, 1987.

[9] R. KOHAVI. *A study of cross-validation and bootstrap for accuracy estimation and model selection.* IJCAI 1995, pp. 1137-1143.

[10] H. W. MEWES, K. HEUMANN, A. KAPS, K. MAYER, F. PFEIFFER, S. STOCKER, D. FRISHMAN *MIPS: a database for protein sequences and complete genomes.* Nucleic Acids Research 27, pp. 44-48, 1999.

[11] G. S. MICHAELS, D. B. CARR, M. ASKENAZI, S. FUHRMAN, X. WEN, R. SOMOGYI. *Cluster analysis and data visualization of large-scale gene expression data.* Pacific Symposium on Biocomputing 3, pp. 42-53, 1998.

[12] G. W. MILLIGAN. *Clustering validation: results and implications for applied analysis.* Clustering and Classification, 1996.

[13] P. TAMAYO, D. SLONIM, J. MESIROV, Q. ZHU, S. KITAREEWAN, E. DMITROVSKY, E. S. LANDER, T. R. GOLUB. *Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation.* PNAS, vol 96, pp. 2907-2912, March 1999.

[14] X. WEN, S. FUHRMAN, G. S. MICHAELS, D. B. CARR, S. SMITH, J. L. BARKER, R. SOMOGYI. *Large-scale temporal gene expression mapping of central nervous system development.* PNAS, vol 95, pp. 334-339, Jan 1998.

Web Sites of data sources

Rat CNS data set: http://rsb.info.nih.gov/mol-physiol/PNAS/GEMtable.html
Yeast cell cycle data set: http://genomics.stanford.edu/yeast/cellcycle.html
Human hematopoietic differentiation data set: http://www.genome.wi.mit.edu/MPR/