DICLENS: Divisive Clustering Ensemble with Automatic Cluster Number

Selim Mimaroglu and Emin Aksehirli

Abstract—Clustering has a long and rich history in a variety of scientific fields. Finding natural groupings of a data set is a hard task as attested by hundreds of clustering algorithms in the literature. Each clustering technique makes some assumptions about the underlying data set. If the assumptions hold, good clusterings can be expected. It is hard, in some cases impossible, to satisfy all the assumptions. Therefore, it is beneficial to apply different clustering methods on the same data set, or the same method with varying input parameters or both. We propose a novel method, DICLENS, which combines a set of clusterings into a final clustering having better overall quality. Our method produces the final clustering automatically and does not take any input parameters, a feature missing in many existing algorithms. Extensive experimental studies on real, artificial, and gene expression data sets demonstrate that DICLENS produces very good quality clusterings in a short amount of time. DICLENS implementation runs on standard personal computers by being scalable, and by consuming very little memory and CPU.

Index Terms—Clustering, combining multiple clusterings, cluster ensembles, consensus clustering, evidence accumulation, minimum spanning tree, gene expressions.

1 Introduction

Clustering is commonly used for discovering the hidden relations and interactions between chemical compounds, genes, and cellular structures. Unsupervised clustering algorithms are used to categorize the experimental results. Moreover, utilization of machine learning methods to predict the possible outcomes is essential for costly experiments, such as microarray experiments.

Microarray experiment, thus clustering, is used for finding the unknown gene functions as well as existence of subtypes of diseases. The categorization is performed on genes in the first case, and it is performed on samples in the latter.

Selecting the best clustering method with the correct parameter values is not possible in most cases. Relatively simpler clustering methods such as k-means and agnes (agglomerative nesting) are commonly used because of their simplicity [1], [2]. However, challenging characteristics of gene expression data sets such as high dimensionality and noise can easily degrade the output quality of even the most sophisticated clustering methods. Therefore, on a gene expression data set several clusterings can be generated and these clusterings can be supplied as input to a clustering ensemble method which produces better quality final clustering [3], [4]. Multiple input clusterings are generated on a data set by

- applying different clustering methods [3],
- running a random initialization clustering method several times [5],
- changing the input parameter values of a clustering method [1], [6], and
- using a subset of the available features, i.e., projection [7].

Recent studies show that clustering ensemble methods produce promising results on gene expression data sets [1], [6], [8], [9], [10]. Most of the existing clustering ensemble methods take input parameters, and their output accuracy is not satisfactory for gene expression data sets. We provide a new clustering ensemble method, DICLENS, which does not take any input parameters. Output of DICLENS is better than the other state-of-the-art methods in terms of accuracy. DICLENS computes the relations between input clusters and displays this information in a tree structure, which can be useful to the laboratory biologists for in depth analysis. Because DICLENS works on cluster level, it is very scalable and runs efficiently on even standard personal computers. Even data sets having huge number of objects can be processed in practical run times. DICLENS is easily available to the biologists and researchers with no coding skills due to its friendly graphical user interface.

DICLENS creates a minimum spanning tree, where each vertex represents an input cluster and each edge shows the intercluster similarity (ECS) between incident vertices. By working on the minimum spanning tree, DICLENS computes an output final clustering having the highest possible clustering quality. Details of DICLENS are presented in Section 3. We define the clustering ensemble problem in the following section and the related work in Section 2. Experimental evaluations are given in Section 4, and the final section has our conclusions.

1.1 Problem Definition

Let D be a data set. A clustering of D, $\pi_i(D)$, can be stated as follows:

$$\pi_i(D) = \{C_{i1}, C_{i2}, \ldots, C_{i|\pi_i(D)|}\},$$

- *The authors are with the Department of Computer Engineering, Bahcesehir University, Ciragan Caddesi, 34353 Besiktas, Istanbul, Turkey. E-mail: {selim.mimaroglu, emin.aksehirli}@bahcesehir.edu.tr.*

Manuscript received 6 Nov. 2010; revised 14 Aug. 2011; accepted 21 Aug. 2011; published online 27 Sept. 2011.
For information on obtaining reprints of this article, please send e-mail to: tcbb@computer.org, and reference IEEECS Log Number TCBB-2010-11-0246. Digital Object Identifier no. 10.1109/TCBB.2011.129.

where C_{ij} is a cluster of $\pi_i(D)$, $1 \leq j \leq |\pi_i(D)|$, and

$$D = \bigcup_{j=1}^{|\pi_i(D)|} C_{ij}.$$

Note that we have a partial clustering (i.e., not complete) when $\bigcup_{j=1}^{|\pi_i(D)|} C_{ij} \subset D$. Given a set of clusterings

$$\Pi(D) = \{\pi_1(D), \pi_2(D), \ldots, \pi_m(D)\},$$

the clustering ensemble problem is defined as finding a new and better clustering $\pi^\star(D) = \{C_1^\star, C_2^\star, \ldots, C_{|\pi^\star(D)|}^\star\}$ by using the information provided in $\Pi(D)$.

2 Related Work

We introduce some of the related work and their corresponding weaknesses for easy evaluation of our new method.

2.1 Combining Multiple Clusterings Using Evidence Accumulation (EAC)

Evidence accumulation [5] accumulates the evidence in each cluster to form a coassociation matrix, SM. Each entry in this matrix, SM_{ij}, is the number of times that objects i and j are assigned to the same clusters. The similarity matrix is provided as input to an agglomerative clustering algorithm, as shown in Algorithm 1.

Algorithm 1. Evidence Accumulation **EAC**
Input: $\Pi(D)$: Multiple Clusterings, n: Number of Objects
Output: $\pi^*(D)$: Final Clustering
Initialize SM as an $n \times n$ matrix
// Construct similarity matrix
for all $d_i \in D$ **do**
  for all $d_j \in D$ **do**
    for all $\pi_k(D) \in \Pi(D)$ **do**
      for all $C_{kl} \in \pi_k(D)$ **do**
        if $d_i \in C_{kl} \wedge d_j \in C_{kl}$ **then**
          $SM_{ij} := SM_{ij} + 1$
Run Agglomerative Clustering on SM to construct $\pi^*(D)$
return $\pi^\star(D)$

2.2 COMUSA

COMUSA [4] is a recent work for combining multiple clusterings into a final clustering, which is a graph-based method producing good results. COMUSA uses the evidence accumulated in the input clusterings, and produces a very good quality final clustering, where the number of clusters in the final clustering is obtained automatically with respect to relaxation rate.

2.3 Cluster-Based Similarity Partitioning Algorithm (CSPA)

CSPA, which is introduced in [3], is based on a coassociation matrix and METIS, which is a software package for partitioning unstructured graphs and hypergraphs [11], [12]. CSPA is shown in Algorithm 2.

Algorithm 2. Cluster-Based Similarity Partitioning Algorithm **CSPA**
Input: $\Pi(D)$: Multiple Clusterings
  n: Number of Objects
  k: Number of final clusters
Output: $\pi^*(D)$: Final Clustering
Initialize SM as an $n \times n$ matrix
// Construct similarity matrix
for all $d_i \in D$ **do**
  for all $d_j \in D$ **do**
    for all $\pi_k(D) \in \Pi(D)$ **do**
      for all $C_{kl} \in \pi_k(D)$ **do**
        if $d_i \in C_{kl} \wedge d_j \in C_{kl}$ **then**
          $SM_{ij} := SM_{ij} + 1$
$\pi^\star(D) := \mathbf{METIS}(SM, k)$
return $\pi^\star(D)$

2.4 Hypergraph Partitioning Algorithm (HGPA)

HGPA is introduced in [3] as well: multiple clusterings construct a hypergraph where each object is a vertex, and each cluster is a hyperedge. Main idea is to have k unconnected components of the hypergraph by using HMETIS [12]. Combining multiple clusterings problem is formulated as partitioning the hypergraph by cutting a minimal number of hyperedges. A set of hyperedges are removed and k unconnected components are obtained, which provides the final clustering.

2.5 Metaclustering Algorithm (MCLA)

In MCLA [3] each cluster is represented by a node in the graph, and edge weights are the Jaccard similarities between corresponding clusters. MCLA is composed of the following steps: 1) constructing the metagraph, 2) partitioning the metagraph using METIS, and 3) computing cluster members, which are shown in Algorithm 3.

Algorithm 3. Meta-Clustering Algorithm **MCLA**
Input: $\Pi(D)$: Multiple Clusterings
  k: Number of Clusters In the Final Clustering
Output: $\pi^\star(D)$: Final Clustering
$G := (E, V, W)$ // Initialize Weighted Meta-Graph
$V := \emptyset$ // Set of Vertices
$E := \emptyset$ // Set of Edges
$W := \emptyset$, $W \subseteq E \rightarrow \mathbb{R}$ // Weight *Function* of Edges
for all $\pi_i(D) \in \Pi(D)$ **do**
  for all $C_{ij} \in \pi_i(D)$ **do**
    $V := V \cup C_{ij}$
for all $C_{ik} \in V$ **do**
  for all $C_{jl} \in V, i \neq j, k \neq l$ **do**
    $E := E \cup (C_{ik}, C_{jl})$
    $W := W \cup (E, Jaccard(C_{ik}, C_{jl}))$
$\pi^\star(D) = \mathbf{METIS}(G, k)$
// Do majority voting
for all $d_i \in D$ **do**
    Assign d_i to its most associated meta-cluster in $\pi^\star(D)$
return $\pi^\star(D)$

2.6 Link-Based Cluster Ensemble (LCE)

LCE [1] constructs a weighted bipartite graph of objects and clusters and partitions the bipartite graph by spectral

clustering. LCE is designed to work on gene expression data sets, and the results in [1] and our experimental evaluations both show that LCE produces good results.

2.7 Other Methods

Another graph-based cluster ensemble method (GCC) that works on microarray data sets is introduced in [10]. GCC computes the number of final clusters automatically when provided with an upper limit. Dudoit and Fridlyand [13] propose two cluster ensemble methods that are based on bagging technique in prediction. Some other good and interesting work on this topic can be found in [6], [14], [15], [16], and [17].

2.8 Weaknesses of Related Work

CSPA, HGPA, MCLA, EAC, and LCE require the number of final clusters in advance. EAC, CSPA, and COMUSA do not scale very well, because they all work at object level. These techniques may not accurately capture the relationship between clusters, which is another disadvantage. Although HGPA is very fast, it is not very accurate due to the degenerative effect of noise clusters. MCLA uses Jaccard measure, which only captures syntactical similarity between clusters. LCE starts with a bipartite membership graph of objects and clusters. But, LCE builds up a dense graph with implied similarities between every cluster and every object which needs a lot of computation.

COMUSA also finds the true number of final clusters but only when it is supplied with the true relaxation parameter.

Although median partition methods implicitly estimate the number of clusters, median partition is a NP-complete [18] problem. Therefore, these methods suffer from slow execution times.

Genetic methods suffer from long execution times as well. In the domain of clustering ensemble, determining chromosome encoding, crossover, mutation, and the fitness function are not immediate.

3 DICLENS

DICLENS approaches the clustering ensemble problem by organizing all the input clusters in the form of minimum spanning tree, where each vertex represents an input cluster, and each edge shows the intercluster similarity between incident vertices. Intercluster similarity between a pair of clusters is an unsupervised objective measure which is defined as follows:

$$ECS(C_{ik}, C_{jl}) = \frac{1}{|C_{ik}||C_{jl}|} \sum_{d \in C_{ik}, d' \in C_{jl}} sim(d, d'). \quad (1)$$

In (1), $sim(d, d')$ is the number of times that objects d and d' are assigned to the same clusters, which is computed from $\Pi(D)$ and is known as *evidence accumulation* in the literature. For example in Fig. 1, ECS between C_{ik} and C_{jl} is

$$ECS(C_{ik}, C_{jl}) = \frac{1}{4}\{sim(d_1, d_3) + sim(d_1, d_4) + sim(d_2, d_3) + sim(d_2, d_4)\}.$$

Large values of ECS indicate close similarities between clusters. In DICLENS, we are aiming to obtain well-separated clusters in the final clustering.

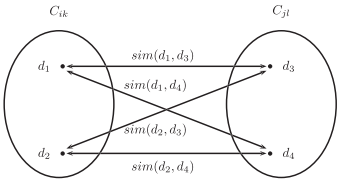


Fig. 1. Intercluster similarity.

DICLENS is shown in Algorithm 4. Input clusterings are converted into an undirected, weighted graph G as shown in Lines 1-11. As expected, in this graph each vertex represents a cluster, and each edge weight represents the ECS value between the corresponding clusters. In order to obtain similarity-based minimum-cost spanning tree (SMST), we run Prim's Algorithm on G as shown in Line 12. But, note that cost of edges and edge weight values W have inverse relationship, since low values of similarity indicate high values of dissimilarity (cost). We explain the core of DICLENS, `find_best_clustering` procedure (Line 13), in the next section.

Algorithm 4. DICLENS: **DI**visive **CL**ustering **ENS**emble with Automatic Cluster Number
Input: $\Pi(D)$: Input Clusterings
Output: $\pi^\star(D)$: Final Clustering
1: $G := (E, V, W)$ // Initialize Weighted Graph
2: $V := \emptyset$ // Set of Vertices
3: $E := \emptyset$ // Set of Edges
4: $W := \emptyset$, $W \subseteq E \rightarrow \mathbb{R}$ // Weight *Function* of Edges
5: **for all** $\pi_i(D) \in \Pi(D)$ **do**
6: **for all** $C_{ij} \in \pi_i$ **do**
7: $V = V \cup C_{ij}$
8: **for all** $C_{ik} \in V$ **do**
9: **for all** $C_{jl} \in V, i \neq j, k \neq l$ **do**
10: $E := E \cup (C_{ik}, C_{jl})$
11: $W := W \cup (E, ECS(C_{ik}, C_{jl}))$
// Cost of edges and W values have inverse relationship
12: $SMST :=$ `Prim's_Algorithm`(G)
13: $\pi^\star(D) :=$ `find_best_clustering`$(SMST)$
14: **return** $\pi^\star(D)$

3.1 Finding the Best Clustering Automatically

Intercluster similarity concept can be expanded for a clustering $\pi_i(D)$ as shown in (2).

$$ECS_\pi(\pi_i(D)) = \frac{1}{\binom{|\pi_i(D)|}{2}} \sum_{C_{ik}, C_{il} \in \pi_i(D), k \neq l} ECS(C_{ik}, C_{il}). \quad (2)$$

Large values of $ECS_\pi(\pi_i(D))$ indicate that cluster pairs of $\pi_i(D)$ are very similar, which is not desired. Well-separated clusters are likely to exist for small values of $ECS_\pi(\pi_i(D))$.

Intracluster similarity (ICS) is based on the similarity of object pairs within a cluster, therefore measures the compactness of a cluster. This notion is formulated in (3), and it is pictured in Fig. 2. As mentioned, $sim(d, d')$ is the number of times that objects d and d' are assigned to the same clusters in $\Pi(D)$

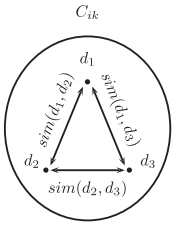


Fig. 2. Intracluster similarity.

$$ICS(C_{il}) = \frac{1}{\binom{|C_{il}|}{2}} \sum_{d,d' \in C_{il}} sim(d, d'). \quad (3)$$

Intracluster similarity concept can be expanded for a clustering $\pi_i(D)$ as shown in (4). Large values of $ICS_\pi(\pi_i(D))$ express that clusters of $\pi_i(D)$ are compact, which is preferred.

$$ICS_\pi(\pi_i(D)) = \frac{1}{|\pi_i(D)|} \sum_{C_{il} \in \pi_i(D)} ICS(C_{il}). \quad (4)$$

Intuitively, compactness and well separateness can be explained as follows: objects in a compact cluster are clustered together many times in the input clusterings. Similarly, objects of two well-separated clusters are rarely clustered together in the input clusterings. Thus, we generate compact and well-separated clusters by utilizing the information in the input.

Quality of a clustering $\pi_i(D)$, is determined by the quality function $\phi(\pi_i(D))$, that is defined as follows:

$$\phi(\pi_i(D)) = \overline{ICS_\pi}(\pi_i(D)) - \overline{ECS_\pi}(\pi_i(D)). \quad (5)$$

In (5), $\overline{ICS_\pi}(\pi_i(D))$ and $\overline{ECS_\pi}(\pi_i(D))$ are min-max normalized versions of $ICS_\pi(\pi_i(D))$ and $ECS_\pi(\pi_i(D))$ as shown in Algorithm 6. By transforming the intercluster and intracluster similarity values into the range of $[0, 1]$ we reduce the adverse effects of large and small values in both measures, since large values dominate the result and small values do not change the outcome.

$ECS_\pi(\pi_i(D))$, $ICS_\pi(\pi_i(D))$, (therefore $\overline{ICS_\pi}(\pi_i(D))$ and $\overline{ECS_\pi}(\pi_i(D))$) can be computed very fast—linear with the number of total clusters in the input clusterings—using very little memory by bit vectors, and binary operations as described in [19] and in Section 3.3.3.

DICLENS tries to find the clustering $\pi^\star(D)$ that maximizes the quality function $\phi(\pi_i(D))$.

$$\pi^\star(D) = \arg\max_{\pi_i(D)} \phi(\pi_i(D)). \quad (6)$$

Algorithm 5 shows the `find_best_clustering` procedure that seeks the best final clustering $\pi^\star(D)$, which has compact and well-separated clusters, by using the quality function $\phi(\pi_i(D))$. Similarity-based minimum-cost spanning tree is constructed for the purpose of finding such a cluster. Starting from the minimum edge weight, `find_best_clustering` cuts an edge in SMST agglomeratively and produces a metaclustering (Line 6)—each metacluster is a connected component of the tree. In SMST, edge weights having the minimum value intuitively indicate weakest links, therefore they are good candidates for removal. After each iteration, connected clusters become more similar to each other, which means objects in that metacluster coexist many times in input clusterings. Because each metacluster contains a set of clusters, an object can occur multiple times in a metacluster. Therefore, majority voting is performed on the metaclustering (Line 7) in order to produce a nonoverlapping final clustering. As a result, each object is assigned to only one cluster where it exists most frequently. Note that, in a SMST there are $n-1$ edges and at most n different clusterings, where n is the number of total clusters in the input clusterings $\Pi(D)$. Among n clusterings, `find_best_clustering` procedure outputs the final clustering having the most compact and most separated clusters—that is the one which produces maximal value in (5). In some cases, there may be more than one clustering producing the maximal value.

Algorithm 5. `find_best_clustering`($SMST$)
Input: $SMST$: Similarity Based Minimum-cost Spanning Tree
Output: k: Cluster Number
1: initialize empty lists: ics, $nics$, ecs, $necs$, ϕ, π
2: $k := 0$ // Store Edge Number
3: **repeat**
4: $k := k + 1$
5: remove minimum weighted edge from $SMST$
6: $\pi_{MC}(D)$ is the meta-clustering of $SMST$, each connected component is a meta-cluster
7: $\pi_k(D) :=$ majority_voting($\pi_{MC}(D)$)
8: $ics_k := ICS_\pi(\pi_k(D))$
9: $ecs_k := ECS_\pi(\pi_k(D))$
10: **until** there are some edges in $SMST$
11: $nics :=$ `normalize`(ics)
12: $necs :=$ `normalize`(ecs)
13: **for** $i := 1$ to k **do**
14: $\phi_i := ics_i - ecs_i$
15: $max\_index := \max_i(\phi)$
16: **return** π_{max_index}

Algorithm 6. `normalize`(list)
Input: $list$: a list of values
Output: $normalized\_list$: min-max normalized list of values in $list$
1: $max := \max(list)$
2: $min := \min(list)$
3: $normalized\_list :=$ initialize empty list
4: **for** $i := start\_of(list)$ to $end\_of(list)$ **do**
5: $normalized\_list_i := \frac{list_i - min}{max - min}$
6: **return** $normalized\_list$

There are $\frac{1}{k!}\sum_{l=1}^{k}\binom{k}{l}(-1)^{k-l}l^n$ possible clusterings, where k is the number of final clusters and n is the number of objects [3]. This number gets even larger if k is not known in advance, therefore exhaustive search is not a practical option. Although it is not guaranteed that the `find_best_clustering` procedure will produce the globally optimum final clustering, experimental results show that final clusterings produced by our algorithm are very good.

3.2 Toy Problem Demonstration

In binary format, a collection of clusterings is presented in Table 1. Note that the number of clusters is not fixed, furthermore some clusterings are not complete: d_3, d_4, and d_5 are not placed in any cluster at $\pi_4(D)$. Similarity-based

TABLE 1
Input Clusterings on a Data Set D

$\Pi(D)$	Clusters	d_1	d_2	d_3	d_4	d_5	d_6
$\pi_1(D)$	C_{11}	1	1	0	0	0	0
	C_{12}	0	0	1	1	0	0
	C_{13}	0	0	0	0	1	1
$\pi_2(D)$	C_{21}	0	0	0	0	1	1
	C_{22}	0	0	1	1	0	0
	C_{23}	1	1	0	0	0	0
$\pi_3(D)$	C_{31}	1	1	1	0	0	0
	C_{32}	0	0	0	1	1	1
$\pi_4(D)$	C_{41}	1	1	0	0	0	0
	C_{42}	0	0	0	0	0	1

minimum spanning tree of Table 1 is shown in Fig. 3a. Starting from the smallest similarity value on SMST, we can cut the SMST into disjoint components, where each component is a metacluster that will represent an output cluster after majority voting. Each step of DICLENS, and the quality of the final clustering produced after each cut is shown in Table 2a. Best quality clusterings according to the objective measure (5) are obtained in Steps 2 and 3: actually, they are the same exact clusterings. According to this output, DICLENS cuts the lowest valued 2 edges (Step 2) and produces the metaclusters shown in Fig. 3b. Majority voting procedure on the metaclusters are displayed in Table 2b; each object is assigned to only one cluster where it exists most frequently. Output of DICLENS is a final clustering with three clusters as shown in Fig. 3c.

3.3 Discussion of DICLENS

In this section, we discuss some important features of DICLENS.

3.3.1 DICLENS is Based on Solid Foundations

In DICLENS, ECS is built on the coexistence information obtained from the input clusterings. Jaccard, for example, completely ignores valuable information provided by the input clusterings; it entirely relies on the syntactical similarity. Therefore, our ECS measure is very effective which provides very accurate cluster level information by utilizing the information in the object level.

Similarity-based Minimum Spanning Tree has been used for clustering purposes for a long time. SMST is a sparse graph, therefore, the processing complexity is low. SMST also preserves the cluster structure of the underlying data: Every cluster is represented as a subtree of SMST [20].

TABLE 2
DICLENS Steps and Majority Voting

(a)

Step No.	Cut Edge	$\phi(\pi^\star(D))$
1	$C_{12}-C_{32}$	0.37
2	$C_{12}-C_{31}$	0.5
3	$C_{13}-C_{32}$	0.5
4	$C_{12}-C_{22}$	-0.27
5	$C_{11}-C_{31}$	-0.46
6	$C_{13}-C_{42}$	-0.46
7	$C_{13}-C_{21}$	-0.46
8	$C_{11}-C_{23}$	-0.46
9	$C_{11}-C_{41}$	-0.46

(b)

Meta-Cluster	d_1	d_2	d_3	d_4	d_5	d_6
$C_1^\star$	4	4	1	0	0	0
$C_2^\star$	0	0	2	2	0	0
$C_3^\star$	0	0	0	1	3	4

Final clusters produced by DICLENS are well separated and compact. This is achieved by combining two very well-known unsupervised objective measures: intracluster and intercluster similarities.

3.3.2 DICLENS Automatically Finds the Number of Clusters

DICLENS does not take any input parameters; a collection of clusterings, $\Pi(D)$, is the only input. Our algorithm works very well with arbitrary number of clusterings and clusters. Most of the clustering and combining multiple clustering algorithms require the number of final clusters in advance. However, DICLENS automatically computes the number of clusters in the final clustering, which is a considerable advantage.

3.3.3 DICLENS Scales Very Well

Using the formulation from [19], which is given in (7) for completeness, intercluster similarity between two clusters can be computed in linear time with a complexity of $O(n)$, where n is the number of clusters in $\Pi(D)$ and in most of the cases $n \ll |D|$, where $|D|$ is the number of objects

$$ECS(C_{\star k}, C_{\star l}) = \frac{1}{|C_{\star k}||C_{\star l}|}\sum_{i=1}^{|\Pi|}\sum_{j=1}^{|\pi_i|}\binom{|(C_{\star k}\vee C_{\star l})\wedge C_{ij}|}{2} - \binom{|C_{\star k}\wedge C_{ij}|}{2} - \binom{|C_{\star l}\wedge C_{ij}|}{2}. \quad (7)$$

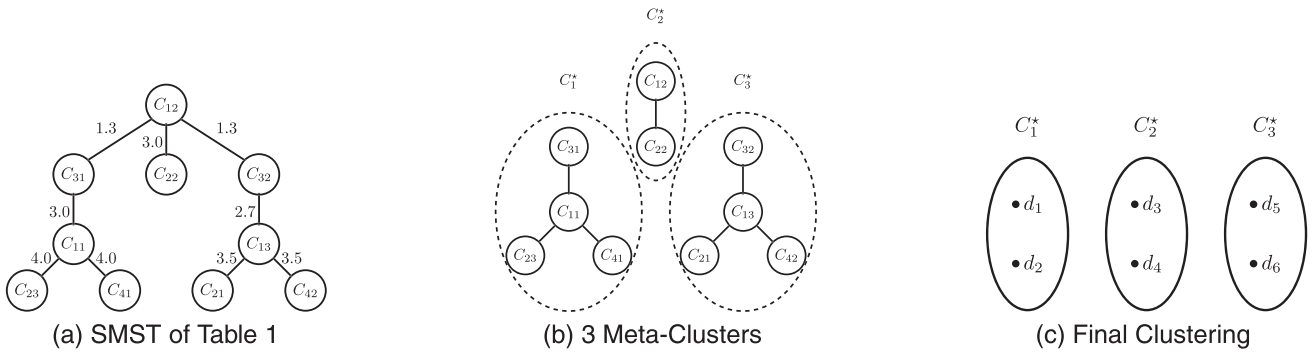


Fig. 3. Toy problem demonstration of DICLENS.

TABLE 3
Abbreviations for ARI

Class \ Cluster	$C^\star_1$	$C^\star_2$	...	$C^\star_p$	Sums
C^o_1	n_{11}	n_{12}	...	n_{1p}	$n_{1.}$
C^o_2	n_{21}	n_{22}	...	n_{2p}	$n_{2.}$
$\vdots$	$\vdots$	$\vdots$		$\vdots$	
C^o_r	n_{r1}	n_{r2}	...	n_{rp}	$n_{r.}$
Sums	$n_{.1}$	$n_{.2}$		$n_{.p}$	$n_{..} = n$

In (7) each cluster is represented as a bit vector where the value of ith bit is 1 if that cluster contains the ith item. Note that the above formulation of $ECS(C_{\star k}, C_{\star l})$ is independent from the number of objects.

4 Experimental Evaluations

In this section, we present an objective cluster validity measure, test data sets, methods for generating input clusterings, and experimental results.

4.1 Measuring Clustering Validity

Quality of clusterings can be evaluated by supervised or unsupervised methods. Because we already have the real class labels of the data sets, we use one of the most widely used supervised evaluation method: adjusted rand index (ARI) [21], which is an improved version of Rand Index [22]. We use ARI for comparing quality of both input clusterings and the clusterings that are obtained by clustering ensemble algorithms with the real class labels. Given a clustering $\pi^\star(D) = \{C^\star_1, C^\star_2, \ldots, C^\star_{|\pi^\star(D)|}\}$ and an original (having real class labels) clustering $\pi^o(D) = \{C^o_1, C^o_2, \ldots, C^o_{|\pi^o(D)|}\}$, where $C^\star_i \cap C^\star_j = \emptyset$ for $1 \le i, j \le |\pi^\star(D)|$, and $C^o_i \cap C^o_j = \emptyset$ for $1 \le i, j \le |\pi^o(D)|$ with variables in Table 3 referring to

$$p = |\pi^\star(D)|, \quad r = |\pi^o(D)|, \quad n_{ij} = |C^o_i \cap C^\star_j|$$

$$n_{i.} = \sum_{j=1}^{p} n_{ij}, \quad n_{.j} = \sum_{i=1}^{r} n_{ij}.$$

ARI is formulated as follows:

$$\frac{\sum_{i,j}\binom{n_{ij}}{2} - \left(\sum_i \binom{n_{i.}}{2}\sum_j \binom{n_{.j}}{2}\right)/\binom{n}{2}}{\frac{1}{2}\left(\sum_i \binom{n_{i.}}{2} + \sum_j \binom{n_{.j}}{2}\right) - \left(\sum_i \binom{n_{i.}}{2}\sum_j \binom{n_{.j}}{2}\right)/\binom{n}{2}}.$$

ARI takes maximum value at 1 which indicates perfect match between two clusterings $\pi^\star(D)$ and $\pi^o(D)$.

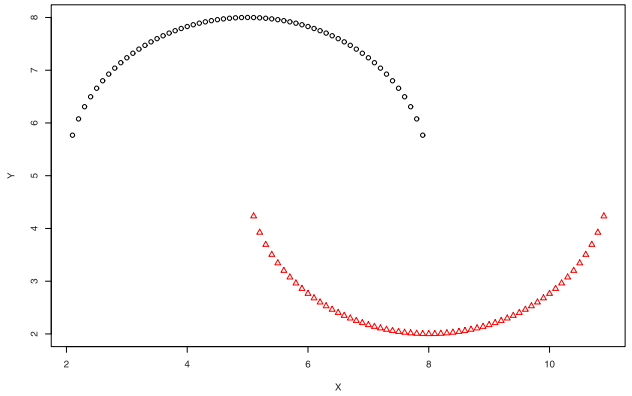
4.2 Data Sets

4.2.1 Gene Expression Data Sets

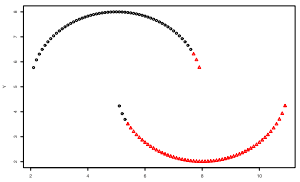
DICLENS is tested on 34 different publicly available gene expression data sets having the properties shown in Table 4.

TABLE 4
Gene Expression Data Sets

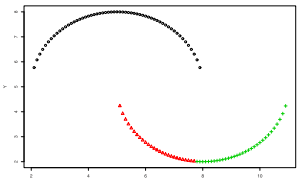
Data Set	*Array Type*	*Tissue*	*Total samples*	*Num of classes*	*Total Genes*	*Selected # of Genes*
Bladder carcinoma [23]	Affymetrix	Bladder	40	3	7129	1203
Breast Cancer [24]	Affymetrix	Breast	49	2	7129	1198
Breast-Colon tumors [25]	Affymetrix	Breast, Colon	104	2	22283	182
Carcinomas [26]	Affymetrix	Multi-tissue	174	10	12533	1571
Central nervous system-1 [27]	Affymetrix	Brain	34	2	7129	857
Central nervous system-2 [27]	Affymetrix	Brain	42	5	7129	1379
Endometrial cancer [28]	Double Channel	Endometrium	42	4	8872	1771
Glioblastoma multiforme [29]	Double Channel	Brain	37	3	24192	1411
Gliomagenesis [30]	Double Channel	Brain	50	3	41472	1739
Gliomas-1 [31]	Affymetrix	Brain	50	4	12625	1377
Gliomas-2 [31]	Affymetrix	Brain	28	2	12625	1070
Gliomas-3 [31]	Affymetrix	Brain	22	2	12625	1152
Hepatocellular carcinoma [32]	Double Channel	Liver	178	2	22699	85
Leukemia-1 [33]	Affymetrix	Bone Marrow	248	2	12625	2526
Leukemia-2 [33]	Affymetrix	Bone Marrow	248	6	4022	1095
Leukemia-3 [34]	Affymetrix	Blood	72	2	12582	1081
Leukemia-4 [34]	Affymetrix	Blood	72	3	12582	2194
Leukemia-5 [35]	Affymetrix	Bone Marrow	72	2	7129	1877
Leukemia-6 [35]	Affymetrix	Bone Marrow	72	3	7129	1877
Lung tumor-1 [36]	Affymetrix	Lung	203	5	12600	1543
Lung tumor-2 [37]	Double Channel	Lung	66	4	24192	4553
Lymphoma-1 [38]	Double Channel	Blood	42	2	4022	1095
Lymphoma-2 [38]	Double Channel	Blood	62	3	4022	2093
Lymphoma-3 [39]	Affymetrix	Blood	77	2	7129	798
Melanoma [40]	Double Channel	Skin	38	2	8067	2201
Mesothelioma [41]	Affymetrix	Lung	181	2	12533	1626
Multi-tissue [42]	Affymetrix	Multi-tissue	190	14	16063	1363
Prostate cancer-1 [43]	Double Channel	Prostate	104	5	20000	2315
Prostate cancer-2 [43]	Double Channel	Prostate	92	4	20000	1288
Prostate cancer-3 [44]	Double Channel	Prostate	69	3	42640	1625
Prostate cancer-4 [44]	Double Channel	Prostate	110	4	42640	2496
Prostate cancer-5 [45]	Affymetrix	Prostate	102	2	12600	339
Round blue-cell tumor [46]	Double Channel	Multi-tissue	83	4	6567	1069
Serrated carcinomas [47]	Affymetrix	Colon	37	2	22883	2202



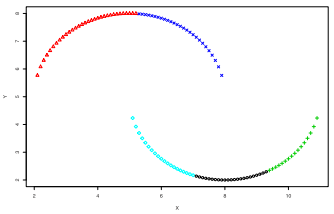
(a) DICLENS Final Clustering, ARI:1.0



(b) Input 1, ARI:0.80

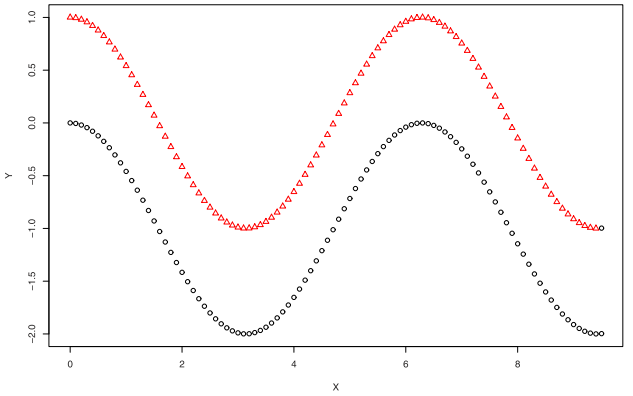


(c) Input 2, ARI:0.75

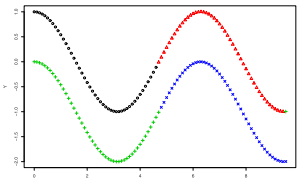


(d) Input 3, ARI:0.41

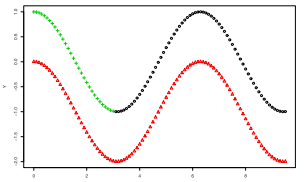
Fig. 4. DICLENS on 2-half rings data set, and three input clusterings.



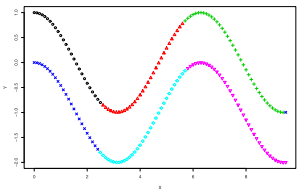
(a) DICLENS Final Clustering, ARI:0.98



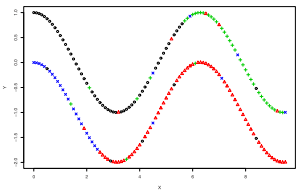
(b) Input 1, ARI:0.49



(c) Input 2, ARI:0.78



(d) Input 3, ARI:0.33



(e) Input 4, ARI: 0.38

Fig. 5. DICLENS on 2-curve data set, and four input clusterings.

Fourteen of the data sets were obtained using Affymetrix chips and 20 of them obtained using double-channel cDNA technology. All of the gene expression data sets are related to cancer research. As described in [2], these data sets are filtered, therefore they do not contain any uninformative genes. Generation of input clusterings from the real-valued data sets is described in detail in Section 4.2.3.

4.2.2 Nonbiological Data Sets

Glass Identification (Glass), and Image Segmentation (Imageseg) data sets are obtained from the University of California Irvine Machine Learning Repository [48]. Glass is a multivariate data set having 10 dimensions, 214 objects, and 6 classes. Imageseg is also a multivariate data set, with 19 real type attributes, 2,310 objects, and 7 classes.

2-half rings, and 2-curve data sets are shown in Fig. 4a and Fig. 5a, respectively. Although these data sets are not large in size, they are very hard to cluster correctly.

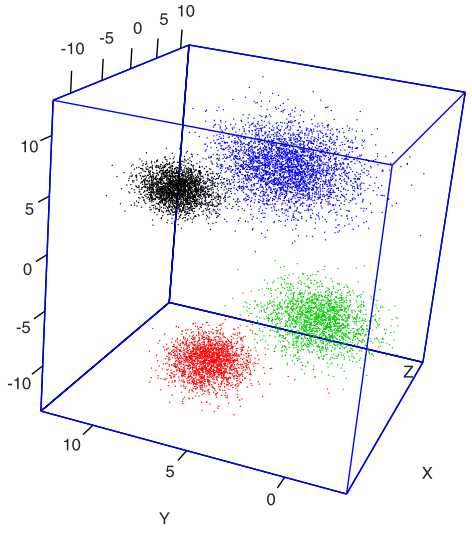
4c10k (Fig. 6a), 4c20k, and 4c40k data sets have 10,000, 20,000, and 40,000 objects, respectively, and they are generated synthetically to form four different Gaussian distributions that are not linearly separable.

4.2.3 Generating Input Clusterings

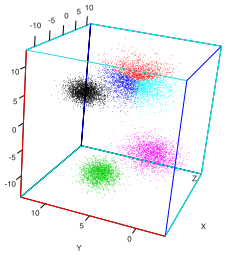
For experimental evaluations, we use different approaches for generating input clusterings: k-means algorithm with varying k-values and random subspacing on gene expression data sets, manually constructing clusters, randomly injecting error into the original clusters, Chameleon [49], and hierarchical agglomerative clustering (agnes) with varying values. Input clusterings generated on the test data sets are shown in Tables 5 and 6. We mostly relied on k-means, remaining methods are used rarely. Note that the diversity and quality of input clusterings impact the quality of the final clustering. We use different approaches and randomly select k values (between 2 and $\sqrt{|D|}$) in k-means algorithm for generating input clusterings, because we expect them to produce a diverse set of clusterings with different properties and qualities. Table 5 shows the properties of input clusterings on gene expression data sets. In this table, Features column indicates the amount of random subspacing, $|\pi|$ is the number of clusters, and $|\Pi|$ stands for the number of clusterings. Last three columns show the min, max, and average quality of input clusterings. Properties of the input clusterings on all the other data sets are shown in Table 6.

4.3 Test Results

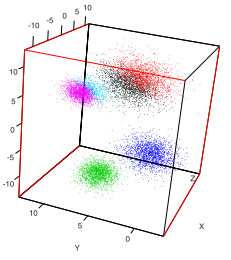
We have conducted experiments on a computer having Intel Centrino Duo 1.67 GHz processor with 1.5 GB of main memory, running on Linux kernel 2.6. Our choice of implementation language for DICLENS and COMUSA is Java, which provides built-in support for bit vectors, and operations on bit vectors. LCE is implemented in MatLab.



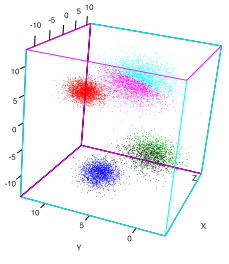
(a) DICLENS Final Clustering, ARI:0.99



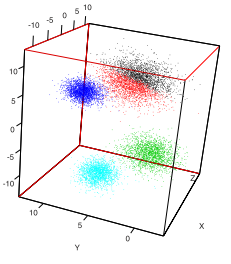
(b) Input 1, ARI:0.76



(c) Input 2, ARI:0.73



(d) Input 3, ARI:0.76



(e) Input 4, ARI:0.82

Fig. 6. DICLENS on 4c10k data set, and four input clusterings.

MCLA, CSPA, and HGPA methods are implemented in Java and C languages. Accuracy of each method is independent of the programming language, however the runtimes may be affected. Java is slower than C, but it is comparable with MatLab in terms of speed. Therefore, better accuracy obtained by DICLENS in short runtime is a very meaningful indicator of superiority.

It is interesting to compare the quality of clusterings produced by DICLENS with the quality of input clusterings. On gene expression data sets and on nonbiological data sets, clustering of DICLENS is always better than the input clusterings with the minimum quality. Similarly, DICLENS always produces better quality clustering than the average quality of all the input clusters on all the data sets. On 70 percent of the gene expression data sets, clustering produced by DICLENS is better than the quality of the input clustering having the maximum quality improvement of DICLENS is in the range of [2 percent, 96 percent] and, on 85.7 percent of nonbiological data sets, clustering produced by DICLENS is better than the quality of the input clustering having the maximum quality, with the improvement in the range of [5 percent, 75 percent]. Extensive experimental results demonstrate that DICLENS can combine the input clusterings into a better quality output clustering effectively. More detailed experimental test results are provided in the following sections.

TABLE 5
Properties of Input Clusterings on Gene Expression Data Sets

Data Set	Method	Features	$\lvert\pi\rvert$	$\lvert\Pi\rvert$	ARI		
					Min	Max	Average
Bladder carcinoma	agnes, k-means	25% - 50%	2 - 6	9	0.18	0.64	0.39
Breast Cancer	k-means	25% - 50%	2 - 7	10	0.08	0.42	0.25
Breast-Colon tumors	k-means	25% - 50%	2 - 10	10	0.11	0.92	0.43
Carcinomas	agnes, k-means	25% - 50%	2 - 13	11	0.10	0.63	0.42
Central nervous system-1	manual	N/A	2 - 4	6	0.21	0.61	0.44
Central nervous system-2	agnes, k-means	25% - 50%	2 - 6	10	0.23	0.54	0.39
Endometrial cancer	manual, random	N/A	4 - 5	5	0.48	0.71	0.60
Glioblastoma multiforme	k-means	75% - 85%	2 - 6	10	-0.03	0.46	0.18
Gliomagenesis	k-means	25% - 50%	2 - 7	10	0.11	0.49	0.28
Gliomas-1	manual	N/A	4 - 6	4	0.48	0.74	0.64
Gliomas-2	manual, random	N/A	2 - 5	4	0.30	0.39	0.36
Gliomas-3	manual	N/A	2 - 3	3	0.37	0.61	0.52
Hepatocellular carcinoma	k-means	75% - 85%	2 - 13	10	0.10	0.70	0.40
Leukemia-1	agnes, k-means	75% - 85%	2 - 15	11	0.10	0.87	0.24
Leukemia-2	k-means	25% - 50%	2 - 15	10	0.14	0.23	0.20
Leukemia-3	manual	N/A	2 - 5	3	0.36	0.56	0.46
Leukemia-4	k-means	75% - 85%	3 - 8	10	0.42	0.92	0.59
Leukemia-5	agnes, k-means	25% - 50%	2 - 8	11	0.15	0.89	0.45
Leukemia-6	k-means	25% - 50%	2 - 8	10	0.18	0.84	0.47
Lung tumor-1	chameleon, k-means	25% - 50%	3 - 14	11	0.10	0.28	0.19
Lung tumor-2	k-means	25% - 50%	2 - 8	10	0.08	0.32	0.19
Lymphoma-1	k-means	25% - 50%	2 - 6	10	0.02	0.43	0.17
Lymphoma-2	k-means	25% - 50%	3 - 7	10	0.20	0.52	0.33
Lymphoma-3	agnes, k-means, random	25% - 50%	2 - 8	10	-0.01	0.32	0.11
Melanoma	manual, random	N/A	2	5	0.38	0.70	0.50
Mesothelioma	k-means	25% - 50%	2 - 13	10	0.07	0.75	0.25
Multi-tissue	chameleon	100%	7 - 14	6	0.06	0.34	0.25
Prostate cancer-1	manual	N/A	5 - 7	5	0.43	0.61	0.52
Prostate cancer-2	manual	N/A	4 - 6	5	0.44	0.61	0.52
Prostate cancer-3	manual	N/A	4 - 7	4	0.24	0.64	0.42
Prostate cancer-4	manual	N/A	5 - 6	3	0.51	0.61	0.55
Prostate cancer-5	k-means	25% - 50%	2 - 10	10	0.02	0.23	0.10
Round blue-cell tumor	agnes, k-means	25% - 50%	2 - 9	9	0.10	0.90	0.49
Serrated carcinomas	manual	N/A	2 - 6	5	0.29	0.51	0.37

TABLE 6
Properties of Input Clusterings on Other Data Sets

Data Set	Method	$\lvert\pi\rvert$	$\lvert\Pi\rvert$	ARI		
				Min	Max	Average
2-curve	manual, random	4 - 6	3	0.33	0.78	0.532
2-half ring	k-means	2 - 5	3	0.414	0.805	0.656
4c10k	k-means	5 - 6	4	0.727	0.818	0.765
4c20k	k-means	3 - 6	9	0.57	0.928	0.739
4c40k	k-means	10	3 - 6	0.557	0.874	0.759
Imageseg	k-means	7	5	0.436	0.525	0.46
Glass	k-means	5 - 7	5	0.581	0.731	0.642

TABLE 7
Quality Results of Final Clusterings on Gene Expression Data Sets

Data Set	DICLENS	MCLA	CSPA	HGPA	LCE	COMUSA
Bladder carcinoma	**0.59**	0.56	0.32	0.36	0.39	0.05
Breast Cancer	**0.63**	0.56	0.44	0.50	0.56	0.23
Breast-Colon tumors	**0.92**	0.85	0.62	0.75	**0.92**	0.39
Carcinomas	0.45	0.47	0.41	0.45	**0.57**	0.15
Central nervous system-1	**1.00**	0.88	0.33	0.33	**1.00**	0.55
Central nervous system-2	0.51	0.51	0.36	0.44	**0.61**	0.27
Endometrial cancer	**1.00**	0.92	0.55	0.42	0.92	0.37
Glioblastoma multiforme	**0.46**	0.16	0.13	0.13	0.16	0.06
Gliomagenesis	**0.55**	0.38	0.25	0.34	0.37	0.13
Gliomas-1	**0.92**	**0.92**	0.73	0.88	**0.92**	0.74
Gliomas-2	0.72	0.60	0.39	0.35	**1.00**	0.32
Gliomas-3	**1.00**	**1.00**	0.51	0.27	**1.00**	0.74
Hepatocellular carcinoma	**0.72**	0.62	0.65	0.02	0.64	0.05
Leukemia-1	**0.96**	0.48	0.10	0.11	**0.96**	0.33
Leukemia-2	**0.38**	0.31	0.25	0.26	0.37	0.01
Leukemia-3	**0.94**	**0.94**	0.44	0.24	0.56	0.15
Leukemia-4	**0.92**	**0.92**	0.81	**0.92**	**0.92**	0.65
Leukemia-5	**0.94**	0.84	0.52	0.33	0.79	0.42
Leukemia-6	**0.84**	0.74	0.54	0.48	0.79	0.05
Lung tumor-1	**0.55**	0.29	0.12	0.11	0.30	0.47
Lung tumor-2	**0.28**	0.25	0.09	0.09	0.15	0.03
Lymphoma-1	**0.50**	0.26	0.37	0.12	0.37	0.02
Lymphoma-2	**0.83**	0.37	0.39	0.35	0.36	0.20
Lymphoma-3	**0.31**	0.20	0.25	0.17	0.25	0.02
Melanoma	**0.89**	**0.89**	**0.89**	0.20	0.70	0.25
Mesothelioma	**0.89**	0.78	0.12	0.14	0.78	0.49
Multi-tissue	0.38	0.32	0.32	0.32	**0.41**	0.39
Prostate cancer-1	**0.89**	**0.89**	0.58	0.52	0.66	0.26
Prostate cancer-2	0.90	**0.92**	0.60	0.47	0.79	0.25
Prostate cancer-3	**0.65**	0.50	0.46	0.27	0.47	0.02
Prostate cancer-4	0.78	0.71	0.44	0.32	**0.86**	0.02
Prostate cancer-5	**0.13**	0.07	0.07	0.05	0.02	0.11
Round blue-cell tumor	**0.94**	0.66	0.49	0.70	0.89	0.27
Serrated carcinomas	**1.00**	0.88	0.20	0.15	**1.00**	0.11

4.3.1 Gene Expression Data Sets

Quality of final clusterings of gene expression data sets, produced by DICLENS, MCLA, CSPA, HGPA, LCE, and COMUSA are shown in Table 7. DICLENS's test results on gene expression data sets are remarkable as shown in the same table. DICLENS identifies four data sets perfectly, and 11 data sets almost perfectly (with ARI values between 0.89 and 0.96). DICLENS is the leader on gene expression data sets, by producing 28 best quality output clusterings on 34 data sets. LCE does well by producing 12 best quality final clusterings. Same table shows that DICLENS never produces very bad clusterings; when it is not leading, its quality measure is close to the best in most cases.

Table 9 shows the execution time results of cluster ensemble methods on gene expression data sets. Although MCLA, CSPA, and HGPA are the fastest methods, they are not as accurate as DICLENS and LCE. COMUSA is fast on data sets that have small number of objects, but its accuracy is not good. Comparing the execution time results of two most accurate methods, that are DICLENS and LCE, DICLENS is remarkably and consistently faster than LCE.

We compare the number of clusters produced by DICLENS with the true number of clusters in Table 11. For 28 data sets out of 34, DICLENS finds the number of clusters perfectly or with just 1 error.

Results of experiments on gene expression data sets show us that cluster ensemble methods provide robustness on the class discovery process. Disadvantages of lack of knowledge about the true number of clusters and informative genes are dramatically reduced when the clusters are combined.

TABLE 8
Quality Results of Final Clusterings on Nonbiological Data Sets

Data Set	DICLENS	MCLA	CSPA	HGPA	LCE	COMUSA
2-curve	**0.98**	**0.98**	**0.98**	0.29	**0.98**	0.28
2-half ring	**1.00**	**1.00**	**1.00**	0.58	**1.00**	0.09
Glass	**1.00**	0.99	0.51	0.21	0.73	0.08
4c10k	**0.99**	0.79	0.68	0	0.98	0.24
4c20k	**0.98**	**0.98**	N/A	0	**0.98**	N/A
4c40k	0.80	**0.98**	N/A	0	**0.98**	N/A
Imageseg	**0.92**	**0.92**	0.91	0.62	0.89	0

TABLE 9
Execution Time Results of Clustering Ensemble Methods on Gene Expression Data Sets (msec)

Data Set	DICLENS	MCLA	CSPA	HGPA	LCE	COMUSA
Bladder carcinoma	59	6	5	59	205	54
Breast Cancer	87	7	7	55	182	103
Breast-Colon tumors	295	7	13	69	369	97
Carcinomas	1139	12	37	249	1533	63
Central nervous system-1	10	5	11	19	67	76
Central nervous system-2	68	7	11	89	340	45
Endometrial cancer	14	7	6	46	103	6
Glioblastoma multiforme	71	6	7	53	169	4
Gliomagenesis	58	10	7	62	297	7
Gliomas-1	28	6	7	61	132	17
Gliomas-2	17	6	12	30	91	3
Gliomas-3	11	7	10	27	50	6
Hepatocellular carcinoma	124	7	22	46	420	57
Leukemia-1	448	8	44	72	476	173
Leukemia-2	2431	16	81	202	2195	132
Leukemia-3	5	7	8	25	78	27
Leukemia-4	207	9	9	124	321	38
Leukemia-5	67	6	8	54	211	13
Leukemia-6	103	7	7	102	268	7
Lung tumor-1	1491	13	31	240	1162	134
Lung tumor-2	69	7	8	82	361	15
Lymphoma-1	124	15	7	36	176	8
Lymphoma-2	134	10	9	81	352	12
Lymphoma-3	162	7	12	55	289	16
Melanoma	2	5	6	11	41	4
Mesothelioma	622	9	25	72	511	101
Multi-tissue	677	10	41	216	3793	97
Prostate cancer-1	38	6	12	79	438	26
Prostate cancer-2	29	6	13	58	187	50
Prostate cancer-3	17	7	8	43	147	53
Prostate cancer-4	29	6	14	48	157	33
Prostate cancer-5	365	10	12	77	463	21
Round blue-cell tumor	112	10	11	93	302	34
Serrated carcinomas	51	6	9	25	72	4

4.3.2 Nonbiological Data Sets

High-dimensional gene expression data sets are not very meaningful to the human eye, therefore we included some challenging nonbiological data sets with low dimensions to demonstrate the effectiveness of DICLENS. For space restrictions, we do not show the graphical representations of all the test data sets. Figs. 4, 5, and 6 represent 2-half rings, 2-curve, 4c10k data sets, and their corresponding input clusterings where natural clusters can be recognized easily. It is clear that although none of the input clusterings correctly identifies the natural clusters, DICLENS can combine the information obtained from the input clusterings and computes the natural clusterings perfectly on 2-half rings, and almost perfectly on 2-curve data, and 4c10k data sets. Similar good results are obtained due to the characteristics of DICLENS on biological and nonbiological data sets.

As shown in Table 8, DICLENS identifies the real clusterings perfectly (with 1.0 ARI values) on 2-half rings and Glass data sets. DICLENS produces almost perfect results on 4c10k, 4c20k, and 2-curve data sets. DICLENS is the leader on seven nonbiological data sets by producing six best quality output clusterings. Output of DICLENS as well

TABLE 10
Execution Time Results of Clustering Ensemble Methods on Nonbiological Data Sets (msec)

Data Set	*DICLENS*	*MCLA*	*CSPA*	*HGPA*	*LCE*	*COMUSA*
2-curve	58	12	29	20	125	91
2-half ring	5	6	19	13	80	71
Glass	84	8	31	55	543	100
4c10k	281	12	36134	551	8585	862288
4c20k	2068	152	N/A	1654	29319	N/A
4c40k	6018	118	N/A	5119	113894	N/A
Imageseg	943	49	5375	200	8876	11977

as the input clusterings provided to DICLENS for 2-half rings, 2-curve, and 4c10k data sets are shown in detail in Figs. 4, 5, and 6. These figures clearly show that the final clusterings produced by DICLENS have better quality than the input clusterings.

Execution time results of DICLENS on nonbiological data sets, shown in Table 10, are similar with the results of gene expression data sets, so that MCLA, CSPA, and HGPA are the fastest ones although they are not very accurate. An interesting observation to note is that when the number of objects increases, execution time of CSPA, COMUSA, and LCE increases very quickly. Moreover, CSPA and COMUSA cannot generate an output clustering for the data sets having more than 10,000 objects. LCE also falls behind in terms of execution time due to memory requirements for large data sets. These results are good demonstrations of possible scalability issues when dealing with gene clustering problems where the number of objects (genes) are very large.

Table 12 shows the number of clusters produced by DICLENS and the real number of classes on nonbiological data sets. DICLENS finds correct number of clusters on all of the nonbiological data sets.

To conclude, results clearly demonstrate that DICLENS produces superior quality final clusterings. We can also conclude that DICLENS produces better results than both MULTI-K and GCC, since in [1] it is shown that LCE produces better quality clusterings than both MULTI-K and GCC.

TABLE 11
Number of Clusters on Gene Expression Data Sets

Data Set	True Cluster #	DICLENS
Bladder carcinoma	**3**	**2**
Breast Cancer	**2**	**2**
Breast-Colon tumors	**2**	**2**
Carcinomas	10	6
Central nervous system-1	**2**	**2**
Central nervous system-2	**5**	**4**
Endometrial cancer	**4**	**4**
Glioblastoma multiforme	**3**	**2**
Gliomagenesis	**3**	**2**
Gliomas-1	**4**	**4**
Gliomas-2	**2**	**2**
Gliomas-3	**2**	**2**
Hepatocellular carcinoma	**2**	**3**
Leukemia-1	**2**	**2**
Leukemia-2	6	3
Leukemia-3	**2**	**2**
Leukemia-4	**3**	**3**
Leukemia-5	**2**	**2**
Leukemia-6	**3**	**3**
Lung tumor-1	5	3
Lung tumor-2	4	2
Lymphoma-1	**2**	**2**
Lymphoma-2	**3**	**2**
Lymphoma-3	**2**	**3**
Melanoma	**2**	**2**
Mesothelioma	**2**	**2**
Multi-tissue	14	5
Prostate cancer-1	**5**	**5**
Prostate cancer-2	**4**	**5**
Prostate cancer-3	**3**	**2**
Prostate cancer-4	**4**	**5**
Prostate cancer-5	2	6
Round blue-cell tumor	**4**	**5**
Serrated carcinomas	**2**	**2**

4.4 A Note on the Validity of the Test Results

For statistical analysis, we performed nonparametric tests by using the software package provided in [50].

Table 13 uses Friedman's method [51] and shows the average rankings of the competing algorithms. It is clear that DICLENS is the winner with the ranking of 1.485, followed by LCE with 2.426. Friedman's measure, χ_F^2, for 5 degrees of freedom is 109.563. Iman and Davenport's measure [52], F_F, with 5 and 165 degrees of freedom, is 59.824. These measures indicate that null hypothesis is

TABLE 12
Number of Clusters on Nonbiological Data Sets

Data Set	True Cluster #	DICLENS
2-curve	**2**	**2**
2-half ring	**2**	**2**
4c10k	**4**	**4**
4c20k	**4**	**4**
4c40k	**4**	**4**
Glass	**6**	**6**
Imageseg	**7**	**7**

TABLE 13
Average Rankings of the Algorithms

Algorithm	Ranking
DICLENS	1.485
LCE	2.426
MCLA	2.720
CSPA	4.324
HGPA	4.750
COMUSA	5.294

TABLE 14
Holm/Hochberg Table for $\alpha = 0.05$

Algorithm	$z = (R_0 - R_i)/SE$	p	Holm/Hochberg/Hommel
COMUSA	8.394	4.689×10^{-17}	0.01
HGPA	7.195	6.243×10^{-13}	0.0125
CSPA	6.255	3.971×10^{-10}	0.0166
MCLA	2.722	0.00647	0.025
LCE	2.0742	0.0380	0.05

TABLE 15
Holm/Hochberg Table for $\alpha = 0.10$

algorithm	$z = (R_0 - R_i)/SE$	p	Holm/Hochberg/Hommel
COMUSA	8.394	4.689×10^{-17}	0.02
HGPA	7.195	6.243×10^{-13}	0.025
CSPA	6.255	3.971×10^{-10}	0.0333
MCLA	2.722	0.00647	0.05
LCE	2.0742	0.0380	0.1

rejected, i.e., ARI values of the competing algorithms are not random, according to critical values in [53, Table B.4].

Since we are comparing DICLENS with other methods, we take the DICLENS results as control group and include the posthoc tests for $1 \times n$ comparisons. p-values of the competing methods are given in Tables 14 and 15.

For $\alpha = 0.05$, Bonferroni-Dunn procedure rejects methods having p-value less than 0.01: COMUSA, HGPA, CSPA, and MCLA. Hochberg procedure rejects methods having p-value less than 0.05: COMUSA, HGPA, CSPA, and MCLA. And finally, Hommel's procedure rejects all of the hypotheses.

For $\alpha = 0.10$, Bonferroni-Dunn procedure rejects methods having p-value less than 0.02: COMUSA, HGPA, CSPA, and MCLA. Hochberg procedure rejects methods having p-value less than 0.1: COMUSA, HGPA, CSPA, and MCLA. And finally, for $\alpha = 0.10$ Hommel's procedure rejects all of the hypotheses.

Results of Tables 14 and 15 clearly show that DICLENS outputs significantly better clusterings than COMUSA, HGPA, CSPA, and MCLA. Although only 1 out of 3 of the posthoc tests confirms that DICLENS produces significantly better results than LCE, average rankings in Table 13 and the quality results in Table 7 indicate that DICLENS is better than LCE as well.

5 Conclusions

Clustering gene expression data efficiently groups together genes having similar functions or tissues having similar characteristics. The study of expression of genes one by one has already provided a wealth of biological insight, but today there exists a variety of methods for creating gene expressions rapidly and efficiently. Immense amount of biological data created by these methods urges the need of automatically organizing and analyzing gene expression data sets that are inherently very challenging.

Clustering methods are used to automatically organize gene data sets such that similar genes are grouped together with respect to their attribute values. Agglomerative clustering methods as well as other techniques such as k-means are widely used for clustering. Similarity of two genes can be measured by a variety of methods such as the euclidean distance, angle or dot product between two vectors. Each clustering method and each similarity measure affect the quality of the clustering. In most cases, it is not possible to know the best clustering method and the best similarity measure which may change with accordance to the data set.

Multiple clusterings of a gene data set by using several clustering methods and several similarity measures can be easily computed due to the high availability of computers. These clusterings can be supplied into a clustering ensemble method such as DICLENS for obtaining a better quality final clustering. Information obtained from the input clusterings raises the quality of the output quality as demonstrated by our experiments.

In this paper, we introduced a novel algorithm for combining a collection of clusterings into a final clustering having better overall quality. Our method, DICLENS, is based on solid foundations since it utilizes minimum-cost spanning tree, intercluster and intracluster similarities for generating clusters that are compact and well separated. DICLENS works on cluster level, therefore scales very well and consumes very little resources. DICLENS does not take any input parameters; it computes the number of clusters automatically. Extensive experimental studies on gene expression data sets demonstrate that DICLENS produces very good quality final clusterings in this domain. DICLENS also works very well on some other real data sets, and on some very challenging artificially generated data sets. All the test data sets, and DICLENS with a GUI is available online at http://www.cs.umb.edu/~smimarog/diclens.

References

[1] N. Iam-on, T. Boongoen, and S. Garrett, "LCE: A Link-Based Cluster Ensemble Method for Improved Gene Expression Data Analysis," *Bioinformatics*, vol. 26, no. 12, pp. 1513-1519, June 2010.

[2] M. de Souto, I. Costa, D. de Araujo, T. Ludermir, and A. Schliep, "Clustering Cancer Gene Expression Data: A Comparative Study," *BMC Bioinformatics*, vol. 9, no. 1, article 497, 2008.

[3] A. Strehl and J. Ghosh, "Cluster—A Knowledge Reuse Framework for Combining Multiple Partitions," *J. Machine Learning Research*, vol. 3, pp. 583-617, Dec. 2002.

[4] S. Mimaroglu and E. Erdil, "Obtaining Better Quality Final Clustering by Merging a Collection of Clusterings," *Bioinformatics*, vol. 26, pp. 2645-2646, 2010.

[5] A. Fred and A. Jain, "Combining Multiple Clusterings Using Evidence Accumulation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 27, no. 6, pp. 835-850, June 2005.

[6] E. Kim, S. Kim, D. Ashlock, and D. Nam, "MULTI-K: Accurate Classification of Microarray Subtypes Using Ensemble K-Means Clustering," *BMC Bioinformatics*, vol. 10, no. 1, article 260, 2009.

[7] A. Topchy, A.K. Jain, and W. Punch, "Combining Multiple Weak Clusterings," *Proc. IEEE Third Int'l Conf. Data Mining*, pp. 331-338, 2003.

[8] H. Cho and I.S. Dhillon, "Coclustering of Human Cancer Microarrays Using Minimum Sum-Squared Residue Coclustering," *IEEE/ACM Trans. Computational Biology and Bioinformatics*, vol. 5, no. 3, pp. 385-400, July-Sept. 2008.

[9] P. Mahata, "Exploratory Consensus of Hierarchical Clusterings for Melanoma and Breast Cancer," *IEEE/ACM Trans. Computational Biology and Bioinformatics*, vol. 7, no. 1, pp. 138-152, Jan.-Mar. 2010.

[10] Z. Yu, H. Wong, and H. Wang, "Graph-Based Consensus Clustering for Class Discovery from Gene Expression Data," *Bioinformatics*, vol. 23, no. 21, pp. 2888-2896, Nov. 2007.

[11] G. Karypis and V. Kumar, "Multilevel Algorithms for Multi-Constraint Graph Partitioning," *Proc. IEEE/ACM Conf. Supercomputing (SC '98)*, p. 28, 1998.

[12] G. Karypis and V. Kumar, "Multilevel K-Way Hypergraph Partitioning," *Proc. 36th Ann. Design Automation Conf.*, pp. 343-348, 1999.

[13] S. Dudoit and J. Fridlyand, "Bagging to Improve the Accuracy of a Clustering Procedure," *Bioinformatics*, vol. 19, no. 9, pp. 1090-1099, June 2003.

[14] X.Z. Fern and C.E. Brodley, "Solving Cluster Ensemble Problems by Bipartite Graph Partitioning," *Proc. 21st Int'l Conf. Machine Learning,* p. 36, 2004.

[15] H.G. Ayad and M.S. Kamel, "On Voting-Based Consensus of Cluster Ensembles," *Pattern Recognition,* vol. 43, no. 5, pp. 1943-1953, May 2010.

[16] X. Wang, C. Yang, and J. Zhou, "Clustering Aggregation by Probability Accumulation," *Pattern Recognition,* vol. 42, no. 5, pp. 668-675, May 2009.

[17] S. Vega-Pons, J. Correa-Morris, and J. Ruiz-Shulcloper, "Weighted Partition Consensus via Kernels," *Pattern Recognition,* vol. 43, no. 8, pp. 2712-2724, Aug. 2010.

[18] J. Barthelemy and B. Leclerc, "The Median Procedure for Partitions," *Partitioning Data Sets, AMS DIMACS Series in Discrete Math.,* vol. 19, pp. 3-34, 1995.

[19] S. Mimaroglu and A.M. Yagci, "A Binary Method for Fast Computation of Inter and Intra Cluster Similarities for Combining Multiple Clusterings," *Proc. Second Int'l Conf. Interaction Sciences: Information Technology, Culture and Human,* pp. 452-456, 2009.

[20] V. Olman, F. Mao, H. Wu, and Y. Xu, "Parallel Clustering Algorithm for Large Data Sets with Applications in Bioinformatics," *IEEE/ACM Trans. Computational Biology and Bioinformatics,* vol. 6, no. 2, pp. 344-352, Apr.-June 2009.

[21] L. Hubert and P. Arabie, "Comparing Partitions," *J. Classification,* vol. 2, no. 1, pp. 193-218, 1985.

[22] W.M. Rand, "Objective Criteria for the Evaluation of Clustering Methods," *J. Am. Statistical Assoc.,* vol. 66, no. 336, pp. 846-850, Dec. 1971.

[23] L. Dyrskjot, T. Thykjaer, M. Kruh0ffer, J.L. Jensen, N. Marcussen, S. Hamilton-Dutoit, H. Wolf, and T.F. Orntoft, "Identifying Distinct Classes of Bladder Carcinoma Using Microarrays," *Nature Genetics,* vol. 33, no. 1, pp. 90-96, Jan. 2003.

[24] M. West, C. Blanchette, H. Dressman, E. Huang, S. Ishida, R. Spang, H. Zuzan, J.A. Olson, J.R. Marks, and J.R. Nevins, "Predicting the Clinical Status of Human Breast Cancer by Using Gene Expression Profiles," *Proc. Nat'l Academy of Sciences USA,* vol. 98, no. 20, pp. 11462-11467, Sept. 2001.

[25] D. Chowdary, J. Lathrop, J. Skelton, K. Curtin, T. Briggs, Y. Zhang, J. Yu, Y. Wang, and A. Mazumder, "Prognostic Gene Expression Signatures Can be Measured in Tissues Collected in RNAlater Preservative," *J. Molecular Diagnostics,* vol. 8, no. 1, pp. 31-39, Feb. 2006.

[26] A.I. Su, J.B. Welsh, L.M. Sapinoso, S.G. Kern, P. Dimitrov, H. Lapp, P.G. Schultz, S.M. Powell, C.A. Moskaluk, H.F. Frierson, and G.M. Hampton, "Molecular Classification of Human Carcinomas by Use of Gene Expression Signatures," *Cancer Research,* vol. 61, no. 20, pp. 7388-7393, Oct. 2001.

[27] S.L. Pomeroy, P. Tamayo, M. Gaasenbeek, L.M. Sturla, M. Angelo, M.E. McLaughlin, J.Y.H. Kim, L.C. Goumnerova, P.M. Black, C. Lau, J.C. Allen, D. Zagzag, J.M. Olson, T. Curran, C. Wetmore, J.A. Biegel, T. Poggio, S. Mukherjee, R. Rifkin, A. Califano, G. Stolovitzky, D.N. Louis, J.P. Mesirov, E.S. Lander, and T.R. Golub, "Prediction of Central Nervous System Embryonal Tumour Outcome Based on Gene Expression," *Nature,* vol. 415, no. 6870, pp. 436-442, Jan. 2002.

[28] J.I. Risinger, G.L. Maxwell, G.V.R. Chandramouli, A. Jazaeri, O. Aprelikova, T. Patterson, A. Berchuck, and J.C. Barrett, "Microarray Analysis Reveals Distinct Gene Expression Profiles among Different Histologic Types of Endometrial Cancer," *Cancer Research,* vol. 63, no. 1, pp. 6-11, Jan. 2003.

[29] Y. Liang, M. Diehn, N. Watson, A.W. Bollen, K.D. Aldape, M.K. Nicholas, K.R. Lamborn, M.S. Berger, D. Botstein, P.O. Brown, and M.A. Israel, "Gene Expression Profiling Reveals Molecularly and Clinically Distinct Subtypes of Glioblastoma Multiforme," *Proc. Nat'l Academy of Sciences USA,* vol. 102, no. 16, pp. 5814-5819, Apr. 2005.

[30] M. Bredel, C. Bredel, D. Juric, G.R. Harsh, H. Vogel, L.D. Recht, and B.I. Sikic, "Functional Network Analysis Reveals Extended Gliomagenesis Pathway Maps and Three Novel MYC-Interacting Genes in Human Gliomas," *Cancer Research,* vol. 65, no. 19, pp. 8679-8689, Oct. 2005.

[31] C.L. Nutt, D.R. Mani, R.A. Betensky, P. Tamayo, J.G. Cairncross, C. Ladd, U. Pohl, C. Hartmann, M.E. McLaughlin, T.T. Batchelor, P.M. Black, A. von Deimling, S.L. Pomeroy, T.R. Golub, and D.N. Louis, "Gene Expression-Based Classification of Malignant Gliomas Correlates Better with Survival than Histological Classification," *Cancer Research,* vol. 63, no. 7, pp. 1602-1607, Apr. 2003.

[32] X. Chen, S.T. Cheung, S. So, S.T. Fan, C. Barry, J. Higgins, K. Lai, J. Ji, S. Dudoit, I.O.L. Ng, M. van de Rijn, D. Botstein, and P.O. Brown, "Gene Expression Patterns in Human Liver Cancers," *Molecular Biology of the Cell,* vol. 13, no. 6, pp. 1929-1939, 2002.

[33] E. Yeoh, M.E. Ross, S.A. Shurtleff, W.K. Williams, D. Patel, R. Mahfouz, F.G. Behm, S.C. Raimondi, M.V. Relling, A. Patel, C. Cheng, D. Campana, D. Wilkins, X. Zhou, J. Li, H. Liu, C. Pui, W.E. Evans, C. Naeve, L. Wong, and J.R. Downing, "Classification, Subtype Discovery, and Prediction of Outcome in Pediatric Acute Lymphoblastic Leukemia by Gene Expression Profiling," *Cancer Cell,* vol. 1, no. 2, pp. 133-143, Mar. 2002.

[34] S.A. Armstrong, J.E. Staunton, L.B. Silverman, R. Pieters, M.L. den Boer, M.D. Minden, S.E. Sallan, E.S. Lander, T.R. Golub, and S.J. Korsmeyer, "MLL Translocations Specify a Distinct Gene Expression Profile that Distinguishes a Unique Leukemia," *Nature Genetics,* vol. 30, no. 1, pp. 41-47, Jan. 2002.

[35] T.R. Golub, D.K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J.P. Mesirov, H. Coller, M.L. Loh, J.R. Downing, M.A. Caligiuri, C.D. Bloomfield, and E.S. Lander, "Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring," *Science,* vol. 286, no. 5439, pp. 531-537, Oct. 1999.

[36] A. Bhattacharjee, W.G. Richards, J. Staunton, C. Li, S. Monti, P. Vasa, C. Ladd, J. Beheshti, R. Bueno, M. Gillette, M. Loda, G. Weber, E.J. Mark, E.S. Lander, W. Wong, B.E. Johnson, T.R. Golub, D.J. Sugarbaker, and M. Meyerson, "Classification of Human Lung Carcinomas by mRNA Expression Profiling Reveals Distinct Adenocarcinoma Subclasses," *Proc. Nat'l Academy of Sciences USA,* vol. 98, no. 24, pp. 13790-13795, Nov. 2001.

[37] M.E. Garber, O.G. Troyanskaya, K. Schluens, S. Petersen, Z. Thaesler, M. Pacyna-Gengelbach, M. van de Rijn, G.D. Rosen, C.M. Perou, R.I. Whyte, R.B. Altman, P.O. Brown, D. Botstein, and I. Petersen, "Diversity of Gene Expression in Adenocarcinoma of the Lung," *Proc. Nat'l Academy of Sciences USA,* vol. 98, no. 24, pp. 13784-13789, Nov. 2001.

[38] A.A. Alizadeh, M.B. Eisen, R.E. Davis, C. Ma, I.S. Lossos, A. Rosenwald, J.C. Boldrick, H. Sabet, T. Tran, X. Yu, J.I. Powell, L. Yang, G.E. Marti, T. Moore, J. Hudson, L. Lu, D.B. Lewis, R. Tibshirani, G. Sherlock, W.C. Chan, T.C. Greiner, D.D. Weisenburger, J.O. Armitage, R. Warnke, R. Levy, W. Wilson, M.R. Grever, J.C. Byrd, D. Botstein, P.O. Brown, and L.M. Staudt, "Distinct Types of Diffuse Large B-Cell Lymphoma Identified by Gene Expression Profiling," *Nature,* vol. 403, no. 6769, pp. 503-511, Feb. 2000.

[39] M.A. Shipp, K.N. Ross, P. Tamayo, A.P. Weng, J.L. Kutok, R.C.T. Aguiar, M. Gaasenbeek, M. Angelo, M. Reich, G.S. Pinkus, T.S. Ray, M.A. Koval, K.W. Last, A. Norton, T.A. Lister, J. Mesirov, D.S. Neuberg, E.S. Lander, J.C. Aster, and T.R. Golub, "Diffuse Large B-Cell Lymphoma Outcome Prediction by Gene-Expression Profiling and Supervised Machine Learning," *Nature Medicine,* vol. 8, no. 1, pp. 68-74, Jan. 2002.

[40] M. Bittner, P. Meltzer, Y. Chen, Y. Jiang, E. Seftor, M. Hendrix, M. Radmacher, R. Simon, Z. Yakhini, A. Ben-Dor, N. Sampas, E. Dougherty, E. Wang, F. Marincola, C. Gooden, J. Lueders, A. Glatfelter, P. Pollock, J. Carpten, E. Gillanders, D. Leja, K. Dietrich, C. Beaudry, M. Berens, D. Alberts, and V. Sondak, "Molecular Classification of Cutaneous Malignant Melanoma by Gene Expression Profiling," *Nature,* vol. 406, no. 6795, pp. 536-540, Aug. 2000.

[41] G.J. Gordon, R.V. Jensen, L. Hsiao, S.R. Gullans, J.E. Blumenstock, S. Ramaswamy, W.G. Richards, D.J. Sugarbaker, and R. Bueno, "Translation of Microarray Data into Clinically Relevant Cancer Diagnostic Tests Using Gene Expression Ratios in Lung Cancer and Mesothelioma," *Cancer Research,* vol. 62, no. 17, pp. 4963-4967, Sept. 2002.

[42] S. Ramaswamy, P. Tamayo, R. Rifkin, S. Mukherjee, C.H. Yeang, M. Angelo, C. Ladd, M. Reich, E. Latulippe, J.P. Mesirov, T. Poggio, W. Gerald, M. Loda, E.S. Lander, and T.R. Golub, "Multiclass Cancer Diagnosis Using Tumor Gene Expression Signatures," *Proc. Nat'l Academy of Sciences USA,* vol. 98, no. 26, pp. 15149-15154, Dec. 2001.

[43] S.A. Tomlins, R. Mehra, D.R. Rhodes, X. Cao, L. Wang, S.M. Dhanasekaran, S. Kalyana-Sundaram, J.T. Wei, M.A. Rubin, K.J. Pienta, R.B. Shah, and A.M. Chinnaiyan, "Integrative Molecular Concept Modeling of Prostate Cancer Progression," *Nature Genetics,* vol. 39, no. 1, pp. 41-51, Jan. 2007.

[44] J. Lapointe, C. Li, J.P. Higgins, M. van de Rijn, E. Bair, K. Montgomery, M. Ferrari, L. Egevad, W. Rayford, U. Bergerheim, P. Ekman, A.M. DeMarzo, R. Tibshirani, D. Botstein, P.O. Brown, J.D. Brooks, and J.R. Pollack, "Gene Expression Profiling Identifies Clinically Relevant Subtypes of Prostate Cancer," *Proc. Nat'l Academy of Sciences USA,* vol. 101, no. 3, pp. 811-816, Jan. 2004.

[45] D. Singh, P.G. Febbo, K. Ross, D.G. Jackson, J. Manola, C. Ladd, P. Tamayo, A.A. Renshaw, A.V. D'Amico, J.P. Richie, E.S. Lander, M. Loda, P.W. Kantoff, T.R. Golub, and W.R. Sellers, "Gene Expression Correlates of Clinical Prostate Cancer Behavior," *Cancer Cell,* vol. 1, no. 2, pp. 203-209, Mar. 2002.

[46] J. Khan, J.S. Wei, M. Ringner, L.H. Saal, M. Ladanyi, F. Westermann, F. Berthold, M. Schwab, C.R. Antonescu, C. Peterson, and P.S. Meltzer, "Classification and Diagnostic Prediction of Cancers Using Gene Expression Profiling and Artificial Neural Networks," *Nature Medicine,* vol. 7, no. 6, pp. 673-679, June 2001.

[47] P. Laiho, A. Kokko, S. Vanharanta, R. Salovaara, H. Sammalkorpi, H. Jarvinen, J. Mecklin, T.J. Karttunen, K. Tuppurainen, V. Davalos, S. Schwartz, D. Arango, M.J. Makinen, and L.A. Aaltonen, "Serrated Carcinomas form a Subclass of Colorectal Cancer with Distinct Molecular Basis," *Oncogene,* vol. 26, no. 2, pp. 312-320, Jan. 2007.

[48] A. Frank and A. Asuncion, *UCI Machine Learning Repository,* School of Information and Computer Sciences, Univ. of California, 2010.

[49] G. Karypis, E. Han, and V. Kumar, "Chameleon: Hierarchical Clustering Using Dynamic Modeling," *Computer,* vol. 32, no. 8, pp. 68-75, 1999.

[50] S. García and F. Herrera, "An Extension on "Statistical Comparisons of Classifiers Over Multiple Data Sets" for All Pairwise Comparisons," *J. Machine Learning Research,* vol. 9, pp. 2677-2694, http://www.jmlr.org/papers/volume9/garcia08a/garcia08a.pdf, Dec. 2008.

[51] M. Friedman, "The Use of Ranks to Avoid the Assumption of Normality Implicit in the Analysis of Variance," *J. Am. Statistical Assoc.,* vol. 32, no. 200, pp. 675-701, http://www.jstor.org/stable/2279372, Dec. 1937.

[52] R.L. Iman and J.M. Davenport, "Approximations of the Critical Region of the Friedman Statistic," *Comm. in Statistics,* vol. A9, pp. 571-595, 1979.

[53] J.H. Zar, *Biostatistical Analysis,* fifth ed. Prentice Hall, Feb. 2009.

Selim Mimaroglu received the MS and PhD degrees in computer science from the University of Massachusetts Boston. Currently, he is working as an assistant professor in the Department of Computer Engineering at Bahcesehir University. He is the director of Data Mining and Machine Learning Research Group. His research interests include fast and efficient methods in data mining, frequent item set detection, clustering, classification, and bioinformatics. For his research work and contributions he has received numerous awards and fellowships from governments, corporations, and universities. He is a member of the IEEE.

Emin Aksehirli received the BS degree in computer engineering at Istanbul Technical University and the MS degree in computer engineering at Bahcesehir University. His MS thesis was directed by Selim Mimaroglu. Currently, he is working as a research assistant at Bahcesehir University. His research interests include data mining and bioinformatics. He is a student member of the IEEE.