CHAPTER 1

Clustering Methods and Their Uses in Computational Chemistry

Geoff M. Downs and John M. Barnard

Barnard Chemical Information Ltd., 46 Uppergate Road, Stannington, Sheffield S6 6BX, United Kingdom

INTRODUCTION

Clustering is a data analysis technique that, when applied to a set of heterogeneous items, identifies homogeneous subgroups as defined by a given model or measure of similarity. Of the many uses of clustering, a prime motivation for the increasing interest in clustering methods is their use in the selection and design of combinatorial libraries of chemical structures pertinent to pharmaceutical discovery.

One feature of clustering is that the process is unsupervised, that is, there is no predefined grouping that the clustering seeks to reproduce. In contrast to supervised learning, where the task is to establish relationships between given inputs and outputs to enable prediction of the output from new inputs, in unsupervised learning only the inputs are available and the task is to reveal aspects of the underlying distribution of the input data. Clustering is thus complemented by the related supervised process of classification, in which items are assigned labels applied to predefined groups: examples include recursive partitioning, naïve Bayesian analysis, and K nearest-neighbor selection. Clustering is a technique for exploratory data analysis and is used increasingly in preliminary analyses of large data sets of medium and high dimensionality as a method of selection, diversity analysis, and data reduction. This chapter reviews the main clustering methods that are used for analyzing chemical

Reviews in Computational Chemistry, Volume 18
Kenny B. Lipkowitz and Donald B. Boyd, Editors
Wiley-VCH, John Wiley and Sons, Inc. © 2002

data sets and gives examples of their application in pharmaceutical companies. Compared to the other costs of drug discovery, clustering can add significant value at minimal cost. First, we provide an outline of clustering as a discipline and define some of the terminology. Then, we give a brief tutorial on clustering algorithms, review progress in developing the methods, and offer some example applications.

Clustering methodology has been developed and used in a variety of areas including archaeology, astronomy, biology, computer science, electronics, engineering, information science, and medicine. Good, general introductory texts on the topic of clustering include those by Sneath and Sokal,[1] Kaufmann and Rousseeuw,[2] Everitt,[3] and Gordon.[4] The main text that is devoted to clustering of chemical data sets is by Willett,[5] with review articles by Bratchell,[6] Barnard and Downs,[7] and Downs and Willett.[8] The present chapter is a complement and update to the latter article. In a previous volume of this series, Lewis, Pickett, and Clark[9] reviewed the use of diversity analysis techniques in combinatorial library design.

As will be shown in the section on Chemical Applications, the current main uses of clustering for chemical data sets are to find representative subsets from high throughput screening (HTS) and combinatorial chemistry, and to increase the diversity of in-house data sets through selection of additional compounds from other data sets. Methods suitable for compound selection are the main focus of this chapter. The methods must be able to handle large data sets of high-dimensional data. For small, low-dimensional data sets, most clustering methods are applicable, and descriptions in the standard texts and implementations available in standard statistical software packages[10,11] suffice. Implementations designed for use on chemical data sets are available from most of the specialist software vendors,[12–17] the majority of which were reviewed by Warr.[18]

The overall process of clustering involves the following steps:

1. Generate appropriate descriptors for each compound in the data set.
2. Select an appropriate similarity measure.
3. Use an appropriate clustering method to cluster the data set.
4. Analyze the results.

This chapter focuses on step 3. For step 1, descriptors may include property values, biological properties, topological indexes, and structural fragments. The performance of these descriptors and forms of representation have been analyzed by Brown[19] and Brown and Martin.[20,21] Similarity searching for step 2 has been discussed by Downs and Willett;[22] characteristics of various similarity measures have been discussed by Barnard, Downs, and Willett.[23,24] For step 4, little has been published specifically about visualization and analysis of results for chemical data sets. However, most publications that focus on implementing systems that utilize clustering do provide details of how the results were displayed or analyzed.

The terminology associated with clustering is extensive, with many terms used to describe the same thing (reflecting the separate development of clustering methods within a multitude of disciplines). Clusters can be *overlapping* or *nonoverlapping*; if a compound occurs in more than one cluster, the clusters are overlapping. At one extreme, each compound is a member of all clusters to a certain degree. An example of this is *fuzzy* clustering in which the degree of membership of an individual compound is in the range 0 to 1, and the total membership summed across all clusters is normally required to be 1. This scheme contrasts with *crisp* clustering in which each compound's degree of membership in any cluster is either 0 or 1. At the other extreme, is the situation wherein each compound is a member of exactly one cluster, in which case the clusters are said to be nonoverlapping. Intermediate situations sometimes occur, where compounds can be members of several, though not of all, clusters. The majority of clustering methods used on chemical data sets generate crisp, nonoverlapping clusters, because analysis of such clusters is relatively simple.

If a data set is analyzed in an iterative way, such that at each step a pair of clusters is merged or a single cluster is divided, the result is *hierarchical*, with a parent–child relationship being established between clusters at each successive level of the iteration. The successive levels can be visualized using a dendrogram, as shown in Figure 1. Each level of the hierarchy represents a partitioning of the data set into a set of clusters. In contrast, if the data set is analyzed to produce a single partition of the compounds resulting in a set of clusters, the result is then *nonhierarchical*. Note that the term *partitioning*

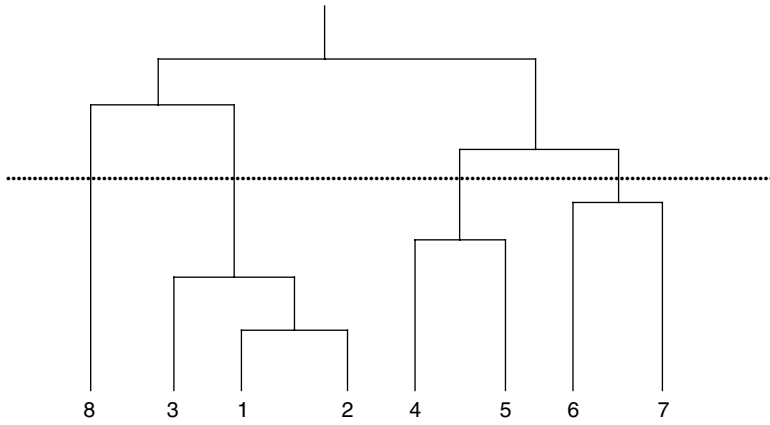

Figure 1 An example of a hierarchy (dendrogram) generated from the clustering of eight items (shown numbered 1–8 across the bottom). The top (root) is a single cluster containing all eight items. The vertical positions of the horizontal lines joining pairs of items or cluster indicate the relative similarities of those pairs. Items 1 and 2 are the most similar and clusters [8,3,1,2] and [4,5,6,7] are the least similar. The dotted horizontal line represents a single partition containing the four clusters [8], [3,1,2], [4,5], and [6,7].

in this context is different from the technique of partitioning (otherwise known as cell-based partitioning). The latter technique is a method of classification rather than of clustering, and a useful review of it, as applied to chemical data sets, is given by Mason and Pickett.[25] A broad classification of the most common clustering methods is shown in Figure 2. Note that, with the wide range of clustering methods devised, some can be placed in more than one of the given categories.

If a hierarchical method starts with all compounds as *singletons* (in clusters by themselves) and the latter are merged iteratively until all compounds are in a single cluster, the method is said to be *agglomerative*. With respect to the dendrogram in Figure 1, it is a bottom-up approach. If the hierarchical method starts with all compounds in a single cluster and iteratively splits one cluster into two until all compounds are singletons, the method is *divisive*, that

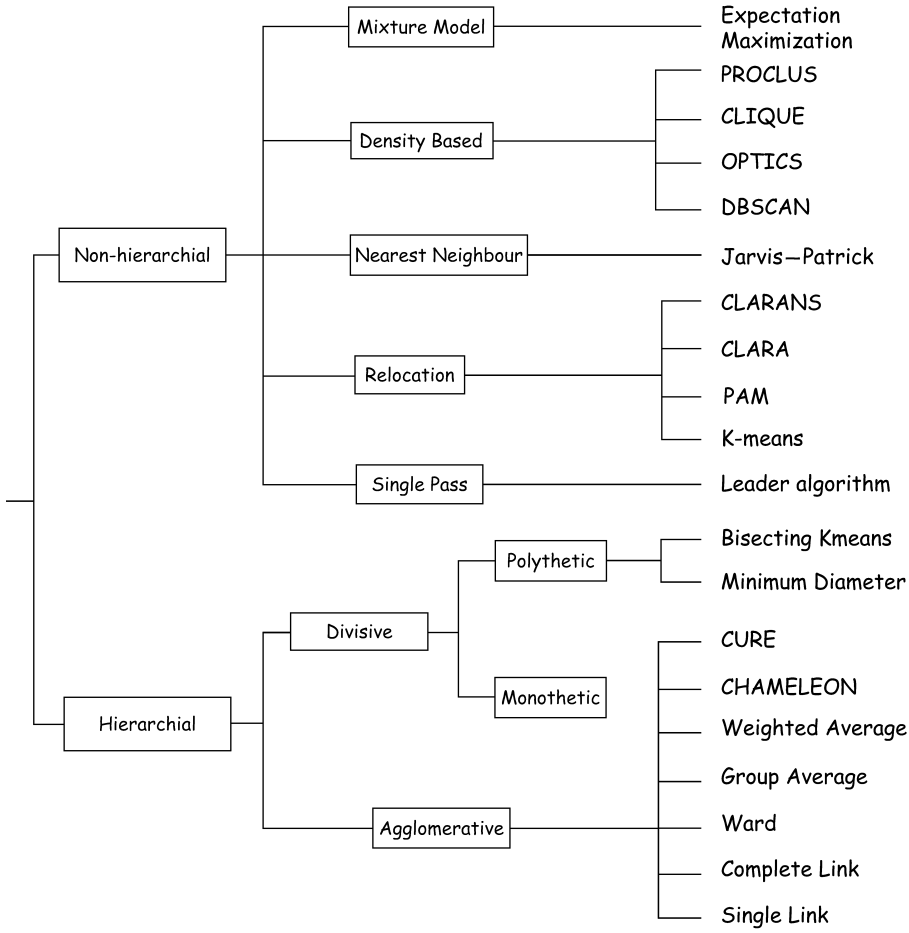

Figure 2 A broad classification of the most common clustering methods.

is, it is a top-down approach. If, at each split, only one descriptor is used to determine how the cluster is split, the method is *monothetic*; otherwise, more descriptors (typically all available) are used, and the method is *polythetic*.

Nonhierarchical methods encompass a wide range of different techniques to build clusters. A *single-pass* method is one in which the partition is created by a single pass through the data set or, if randomly accessed, in which each compound is examined only once to decide which cluster it should be assigned to. A *relocation* method is one in which compounds are moved from one cluster to another to try to improve on the initial estimation of the clusters. The relocating is typically accomplished based on improving a cost function describing the "goodness" of each resultant cluster. The *nearest-neighbor* approach is more compound centered than are the other nonhierarchical methods. In it, the environment around each compound is examined in terms of its most similar neighboring compounds, with commonality between nearest neighbors being used as a criterion for cluster formation. In *mixture model* clustering the data are assumed to exist as a mixture of densities that are usually assumed to be Gaussian (normal) distributions, since their densities are not known in advance. Solutions to the mixture model are derived iteratively in a manner similar to the relocation methods. *Topographic* methods, such as use of Kohonen maps, typically apply a variable cost function with the added restriction that topographic relationships are preserved so that neighboring clusters are close in descriptor space. Other nonhierarchical methods include *density-based* and *probabilistic* methods. Density-based, or mode-seeking, methods regard the distribution of descriptors across the data set as generating patterns of high and low density that, when identified, can be used to separate the compounds into clusters. Probabilistic clustering generates nonoverlapping clusters in which a compound is assigned a probability, in the range 0 to 1, that it belongs to the chosen cluster (in contrast to fuzzy clustering in which the clusters are overlapping and the degree of membership is not a probability).

Having now provided a broad overview of clustering methodology, we next focus on the "classical" methods, which include hierarchical and single-pass, relocation, and nearest-neighbor nonhierarchical techniques. The classification we have described in Figure 2 is one that is commonly used by many scientists; however, it is just one of many possible classifications. Another way to differentiate between clustering techniques is to consider *parametric* and *nonparametric* methods. Parametric methods require distance-based comparisons be made. Here access to the descriptors is required (typically given as Euclidean vectors), rather than just a proximity matrix derived from the descriptors. Parametric methods can be further organized into *generative* and *reconstructive* methods. Generative methods, including mixture model, density-based, and probabilistic techniques, try to match parameters (e.g., cluster centers, variances within and between clusters, and mixing coefficients for the descriptor distributions) to the distribution of descriptors within the data set. Reconstructive methods, such as relocation and topographic, are

based upon improving a given cost function. Nonparametric methods make fewer assumptions about the underlying data; they do not adapt given parameters iteratively and, in general, need only a matrix of pairwise proximities (i.e., a distance matrix).

The term proximity is used here to include similarity and dissimilarity coefficients in addition to distance measures. Individual proximity measures are not defined in this review; full definitions can be found in standard texts and in the articles by Barnard, Downs, and Willett.[23,24] We now define the terms *centroid* and *square-error*, because they will be used throughout this chapter. For a cluster of s compounds each represented by a vector, let $\mathbf{x}(r)$ be the rth vector. The vector of the cluster centroid, $\mathbf{x}(c)$, is then defined as

$$\mathbf{x}(c) = \left(\frac{1}{s}\right) \sum_{r=1}^{s} \mathbf{x}(r) \qquad [1]$$

Note that the centroid is the simple arithmetic mean of the vectors of the cluster members, and this mean is frequently used to represent the cluster as a whole. In situations where a mean is not applicable or appropriate, the median can be used to define the cluster *medoid* (see Kaufman and Rousseeuw[2] for details). The square-error (also called the *within-cluster variance*), e^2, for a cluster is the sum of squared Euclidean distances to the centroid or medoid for all s items in that cluster:

$$e^2 = \sum_{r=1}^{s} [\mathbf{x}(r) - \mathbf{x}(c)]^2 \qquad [2]$$

The square-error across all K clusters in a partition is the sum of the square-errors for each of the K clusters. (Note also that the standard deviation would be the square root of the square-error.)

CLUSTERING ALGORITHMS

This chapter concentrates on the "classical" clustering methods, because they are the methods that have been applied most often in the chemical community. Standard reference works devoted to clustering algorithms include those by Hartigan,[26] Murtagh,[27] and Jain and Dubes.[28]

Hierarchical Methods

Hierarchical Agglomerative

The most commonly implemented hierarchical clustering methods are those belonging to the family of *sequential agglomerative hierarchical non-overlapping* (SAHN) methods. These are traditionally implemented using

what is known as the *stored-matrix* algorithm, so named because the starting point is a matrix of all pairwise proximities between items in the data set to be clustered. Each cluster initially corresponds to an individual item (singleton). As clustering proceeds, each cluster may contain one or more items. Eventually, there evolves one cluster that contains all items. At each iteration, a pair of clusters is merged (agglomerated) and the number of clusters decreases by 1. The stored-matrix algorithm proceeds as follows:

1. Calculate the initial proximity matrix containing the pairwise proximities between all pairs of clusters (singletons) in the data set.
2. Scan the matrix to find the most similar pair of clusters, and merge them into a new cluster (thus replacing the original pair).
3. Update the proximity matrix by inactivating one set of entries of the original pair and updating the other set (now representing the new cluster) with the proximities between the new cluster and all other clusters.
4. Repeat steps 2 and 3 until just one cluster remains.

The various SAHN methods differ in the way in which the proximity between clusters is defined in step 1 and how the merged pair is represented as a single cluster in step 3. The proximity calculation in step 3 typically uses the Lance–Williams matrix-update formula:[29]

$$\mathrm{d}[k,(i,j)] = \alpha_{\mathrm{i}}\,\mathrm{d}[k,i] + \alpha_{\mathrm{j}}\,\mathrm{d}[k,j] + \beta\,\mathrm{d}[i,j] + \gamma\left|\mathrm{d}[k,i] - \mathrm{d}[k,j]\right| \qquad [3]$$

where $\mathrm{d}[k,(i,j)]$ is the proximity between cluster k and the cluster (i, j) formed from merging clusters i and j. Different values for α_{i}, α_{j}, β, and γ define various SAHN methods, some of which are shown in Table 1 and described below.

In *single-link* clustering, the proximity between two clusters is the minimum distance between any pair of items (one from each cluster), that is, the closest pair of points between each cluster. In contrast, in *complete-link* clustering, the proximity between two clusters is the maximum distance between any pair of items, that is, the farthest pair of points between each cluster. Single-link and complete-link represent the extremes of SAHN clustering. In

Table 1 Parameter Values for Some Common SAHN Methods Defined by the Lance–Williams Matrix Update Formula[a]

SAHN Method	α_i	α_j	β	γ
Single-link	0.5	0.5	0	−0.5
Complete-link	0.5	0.5	0	0.5
Group-average	$\frac{N_i}{N_i+N_j}$	$\frac{N_j}{N_i+N_j}$	$\frac{-N_i \times N_j}{(N_i+N_j)^2}$	0
Ward	$\frac{N_i+N_k}{N_i+N_j+N_k}$	$\frac{N_j+N_k}{N_i+N_j+N_k}$	$\frac{-N_k}{N_i+N_j+N_k}$	0

[a] The parameters N_i, N_j, and N_k = number of compounds in clusters i, j, and k, respectively.

the middle is *average-link* clustering in which the proximity between two clusters is the arithmetic average of distances between all pairs of items. Also in the middle is *Ward's* method[30] in which the proximity is the variance between the clusters (where variance is defined as the sum of square-errors of the clusters; see Eq. [2]). At each iteration, the pair of clusters chosen is that whose merger produces the minimum change in square-error (or within-cluster variance; hence the method is also known as the *minimum-variance* method). As the number of clusters decreases, the square-error across all clusters increases. Ward's method minimizes the square-error increase and minimizes the intracluster variance while maximizing the intercluster variance. Because a cluster is represented by its centroid, Ward's method is classified as a *geometric* or *cluster-center* method. Other methods such as the single-link, complete-link, and group-average methods are classified as *graph-theoretic* or *linkage methods*. Murtagh[27] introduced the concept of a *reducibility property* that is applicable to geometric methods. The reducibility property states that for the merger of two clusters, a and b, to form cluster c, there cannot be another cluster, d, that is closer to c than to a or b. If the method satisfies the reducibility property, agglomerations can be performed in localized areas of the proximity space and then amalgamated to produce the full hierarchy. Ward's method, implemented using the Euclidean distance as the proximity measure, is one of the few geometric methods satisfying the reducibility property. Voorhees[31] subsequently showed that if the cosine coefficient of similarity is used as the proximity measure, the group-average method can be implemented as a geometric method, and it satisfies the reducibility property.

For a data set of N compounds, the stored-matrix algorithm for SAHN methods requires $O(N^2)$ time and $O(N^2)$ space for creation and storage of the proximity matrix while requiring $O(N^3)$ time for the clustering. This algorithm is thus very demanding of resources for anything other than small data sets. The importance of the reducibility property is that it enables the stored-matrix algorithm to be replaced by the more efficient *reciprocal nearest-neighbor* (RNN) algorithm that requires only $O(N^2)$ time and $O(N)$ space. Because agglomerations can be performed in localized areas of the proximity space, the RNN algorithm works by tracing paths through proximity space from one point to its nearest neighbor until a point is reached whose nearest neighbor is the previous point in the path, that is, a pair of points that are reciprocal nearest neighbors. These points represent a pair that should be merged into a single point as one of the agglomerations of the full hierarchy. The RNN algorithm is carried out using the following steps:

1. Mark all points as "unused."
2. Begin at an unused point and trace a path of unused nearest neighbors until a reciprocal nearest neighbor pair is found.
3. Add the pair of points to the list of RNNs along with the proximity between them; mark one of the pair of points as "used" (to inactivate it and

its centroid) and replace the centroid of the other point by the centroid of the merged pair.
4. Continue the path tracing from the penultimate point in the path if one exists; otherwise start path tracing from a new, unused starting point.
5. Repeat steps 2–4 until only one unused point remains.
6. Sort the list of RNNs by decreasing proximity values; the sorted list represents the agglomerations needed to construct the hierarchy.

Because path tracing moves from one nearest neighbor to the next, random access to each point is required.

Hierarchical Divisive

Most hierarchical divisive methods are monothetic, meaning that each split is determined on the basis of a single descriptor. The methods differ in how the descriptor is chosen with one possibility being to select the descriptor that maximizes the distance between the resultant clusters. Monothetic divisive methods are usually faster than the SAHN methods described above and have found utility in biological classification. However, for chemical applications, monothetic division often gives poor results when compared to polythetic division or SAHN methods, even though the closely related classification method of recursive partitioning can be very effective in chemical applications (e.g., see the article by Chen, Rusinko, and Young[32]). Unfortunately, most polythetic divisive methods are very resource demanding (more so than for SAHN methods), and accordingly they have not been used much for chemical applications. One exception is the *minimum-diameter* method published by Guenoche, Hansen, and Jaumard;[33] it requires $O(N^2 \log N)$ time and $O(N^2)$ space. This method is based on dividing clusters at each iteration in such a way as to minimize the cluster diameter. The cluster diameter is defined as the largest dissimilarity between any pair of its members, with singleton clusters having a diameter of zero. The minimum-diameter algorithm accomplishes its task by carrying out the following steps:

1. Generate a sorted list of all $N(N-1)/2$ dissimilarities, with the most dissimilar pairs listed first.
2. Perform an initial division by selecting the first pair from the sorted list (i.e., the most dissimilar points in the data set); assign every other point to the closest of the pair.
3. Choose the cluster with the largest diameter and divide it into two clusters so that the larger cluster has the smallest possible diameter.
4. Repeat step 3 for a maximum of $N-1$ divisions.

Nonhierarchical Methods

Single-Pass

Methods that cluster data on the basis of a single scan of the data set are referred to as single-pass. A proximity threshold is typically used to decide

whether a compound is assigned to an existing cluster (represented as a centroid) or if it should be used to start a new cluster. The first compound selected becomes the first cluster; a single sequential scan of the data set then assigns the remaining compounds, and cluster centroids are updated as each compound is assigned to a particular cluster. The most common single-pass algorithm is called the *leader algorithm*. The leader algorithm carries out the following steps to provide a set of nonhierarchical clusters:

1. Set the number of existing clusters to zero.
2. Use the first compound in the data set to start the first cluster.
3. Calculate the similarity, using some appropriate measure, between the next compound and all the existing clusters. If its similarity to the most similar existing cluster exceeds some threshold, assign it to that cluster; otherwise use it to start a new cluster.
4. Repeat step 2 until all compounds have been assigned.

This method is simple to implement and very fast. The major drawback is that it is order dependent; if the compounds are rearranged and scanned in a different order, then the resulting clusters can be different.

Nearest Neighbor

A simple way to isolate dense regions of proximity space is to examine the nearest neighbors of each compound to determine groups with a given number of mutual nearest neighbors. Although several nearest-neighbor methods have been devised, the *Jarvis–Patrick method*[34] is almost exclusively used for chemical applications. The method proceeds in two stages.

The first stage generates a list of the top K nearest neighbors for each of the N compounds, with proximity usually measured by the Euclidean distance or the Tanimoto coefficient;[23] K is typically 16 or 20. The Tanimoto coefficient has been found to perform well for chemical applications where the compounds are represented by fragment screens (bit strings denoting presence/absence of structural features). For finding nearest neighbors with Tanimoto coefficients as a proximity measure, one can use an efficient inverted file approach described by Perry and Willett[35] to speed up the creation of nearest-neighbor lists.

The second stage scans the nearest-neighbor lists to create clusters that fulfill the three following *neighborhood conditions*:

1. Compound i is in the top K nearest-neighbor list of compound j.
2. Compound j is in the top K nearest-neighbor list of compound i.
3. Compounds i and j have at least *Kmin* of their top K nearest neighbors in common, where *Kmin* is user-defined and lies in the range 1 to K.

Pairs of compounds that fail any of the above conditions are not put into the same cluster.

To scan the nearest-neighbor lists and create the clusters in this stage of nonhierarchical clustering, the following three steps are carried out:

1. Tag each compound with a sequential cluster label so that each is a singleton.
2. For each pair of compounds, i and j ($i < j$), compare the nearest-neighbor lists on the basis of the three neighborhood conditions. If the three conditions are passed, replace the cluster label for compound j with the cluster label for compound i. Then, scan all previously processed compounds and replace any occurrences of the cluster label for compound j by the cluster label for compound i.
3. Scan to extract clusters by retrieving all compounds assigned the same cluster label.

The Jarvis–Patrick method requires $O(N^2)$ time and $O(N)$ space.

Relocation

Relocation methods start with an initial guess as to where the centers of clusters are located. The centers are then iteratively refined by shifting compounds between clusters until stability is achieved. The resultant clustering is reliant upon the initial selection of seed compounds that serve as cluster centers. Hence, relocation methods can be adversely affected by outlier compounds. [An *outlier* is a cluster of one item (a singleton or noise). It is on its own, and the clustering method has not put it with anything else because it is not similar enough to anything else.] The iterative refinement seeks an optimal partitioning of the compounds but would likely find a suboptimal solution because it would require the analysis of all possible solutions to guarantee finding the global optimum. Nevertheless, the computational efficiency and mathematical foundation of these methods have made them very popular, especially with statisticians.

The best-known relocation method is the *k-means* method, for which there exist many variants and different algorithms for its implementation. The k-means algorithm minimizes the sum of the squared Euclidean distances between each item in a cluster and the cluster centroid. The basic method used most frequently in chemical applications proceeds as follows:

1. Choose an initial set of k seed compounds to act as initial cluster centers.
2. Assign each compound to its nearest cluster centroid (classification step).
3. Recalculate each cluster centroid (minimization step).
4. Repeat steps 2 and 3 for a given number of iterations or until no compounds are moved from one cluster to another.

In step 1, the initial compounds are usually selected at random from the data set. Random selection is quick and, for large heterogeneous data sets, likely to provide a reasonable initial set. Steps 2 and 3 can be performed separately or in combination. If done separately, the classification (step 2) is performed on

all compounds before recalculation of each cluster centroid (step 3). This approach is referred to as *noncombinatorial* (or *batch update*) classification. If steps 2 and 3 are done in combination, moving a compound from its current cluster to a new cluster (step 2) immediately necessitates recalculation of the affected cluster centroids (step 3). This latter approach is called *combinatorial* or *online update* classification. Most implementations for chemical applications use noncombinatorial classification. In step 4, convergence to a point where no further compounds move between clusters, is usually rapid, but, for safety, a maximum number of iterations can be specified. k-Means clustering requires $O(Nmk)$ time and $O(k)$ space. Here, m is the number of iterations to convergence, and k is the number of clusters. Because m is typically much smaller than N and the effect of k can be reduced substantially through efficient implementation, k-means algorithms essentially require $O(N)$ time.

Mixture Model

Clustering can be viewed as a density estimation problem. The basic premise used in such an estimation is that in addition to the observed variables (i.e., descriptors) for each compound there exists an unobserved variable indicating the cluster membership. The observed variables are assumed to arrive from a mixture model, and the mixture labels (cluster identifiers) are hidden. The task is to find parameters associated with the mixture model that maximize the likelihood of the observed variables given the model. The probability distribution specified by each cluster can take any form. Although mixture model methods have found little use in chemical applications to date, they are mentioned here for completeness and because they are obvious candidates for use in the future.

The most widely used and most effective general technique for estimating the mixture model parameters is the expectation maximization (EM) algorithm.[36] It finds (possibly suboptimally) values of the parameters using an iterative refinement approach similar to that given above for the k-means relocation method. The basic EM method proceeds as follows:

1. Select a model and initialize the model parameters.
2. Assign each compound to the cluster(s) determined by the current model (expectation step).
3. Reestimate the parameters for the current model, given the cluster assignments made in step 2, and generate a new model (maximization step).
4. Repeat steps 2 and 3 for n iterations or until the nth and $(n-1)$th model are sufficiently close.

This method requires prior specification of a model and typically takes a large number of iterations to converge.

Note that the k-means relocation method is really a special case of EM that assumes: (1) each cluster is modeled by a spherical Gaussian distribution, (2) each data item is assigned to a single cluster, and (3) the mixture weights

are equal. Assignment of each compound to the closest-cluster centroid is the expectation step; recalculation of the cluster centroids (model parameters) after assignment is the maximization step.

Topographic

Topographic clustering methods attempt to preserve the proximities between clusters, thus facilitating visualization of the clustering results. For k-means clustering, the cost function is invariant, whereas in topographic clustering it is not, and a predefined neighborhood is imposed on the clusters to preserve the proximities between them. The Kohonen, or self-organizing, map,[37,38] apart from being one of the most commonly used types of neural network, is also a topographic clustering method. A *Kohonen network* uses an unsupervised learning technique to map higher dimensional spaces of a data set down to, typically, two or three dimensions (2D or 3D), so that clusters can be identified from the neurons' coordinates (topological position); the values of the output are ignored. Initially, the neurons are assigned weight vectors with random values (weights). During the self-organization process, the data vectors of the neuron having the most similar weight vector to each data vector and its immediately adjacent neurons are updated iteratively to place them closer to the data vector. The *Kohonen mapping* thus proceeds as follows:

1. Initialize each neuron's weight vector with random values.
2. Assign the next data vector to the neuron having the most similar weight vector.
3. Update the weight vector of the neuron of step 2 to bring it closer to the data vector.
4. Update neighboring weight vectors using a given updating function.
5. Repeat steps 2–4 until all data vectors have been processed.
6. Start again with the first data vector, and repeat steps 2–5 for a given number of cycles.

The iterative adjustment of weight vectors is similar to the iterative refinement of k-means clustering to derive cluster centroids. The main difference is that adjustment affects neighboring weight vectors at the same time. Kohonen mapping requires $O(Nmn)$ time and $O(N)$ space, where m is the number of cycles and n the number of neurons.

Other Nonhierarchical Methods

We have delineated the main categories of clustering methods applicable to chemical applications above. We have also provided one basic algorithm as an example of each. Researchers in other disciplines sometimes use variants of these main categories. The main categories that have been used by those researchers but omitted here include density-based clustering and graph-based clustering techniques. These will be mentioned briefly in the next section.

PROGRESS IN CLUSTERING METHODOLOGY

The representations used for chemical compounds are typically "*data-prints*" (tens or hundreds of real number descriptors, such as topological indexes and physicochemical properties) or *fingerprints* (thousands of binary descriptors indicating the presence or absence of 2D structural fragments or 3D pharmacophores). These numbers can be compared to the tens or hundreds of descriptors typically encountered in data mining and the thousands of descriptors encountered in information retrieval. We now outline the development of clustering methods that are suited to handling these representations and that have been, or in the near future are likely to be, used for chemical applications. Specific examples of chemical applications are given later in the section entitled Chemical Applications.

Algorithm Developments

Having briefly outlined the basic algorithms that are implemented in many of the standard clustering methods, we now set the algorithms in context by reviewing their historical development, discuss the characteristics of each method, and then highlight some of the variants that have been developed for overcoming certain limitations. Clustering is now such a large area of research and everyday use that this chapter must be selective rather than comprehensive in scope. The interested reader can access further details from the references cited throughout this chapter and from the recent review by Murtagh.[39]

Most of the development of hierarchical clustering methods occurred from the 1960s through the mid-1980s, after which there was a period of consolidation, with little new development until recently. From this developmental period, two key publications were those of Lance and Williams[29] in 1967 and the review of hierarchical clustering methods by Murtagh[27] in 1985. Following this developmental period, a few variations have been proposed. Matula[40] developed algorithms that implemented both divisive and agglomerative average-linkage methods, but with high computational costs for processing large data sets. That same year, Jain, Indrayan, and Goel[41] compared single and complete linkage, group and weighted average, centroid, and median agglomerative methods and concluded that complete linkage performed best in failing to find clusters from random data. Podani[42] produced a useful classification of agglomerative methods, in which the standard Lance–Williams update recurrence formula[29] is split into two formulas. He also introduced three new parameter variations, that is, three new agglomerative methods were defined, but these seem to represent more of an inclusion for the sake of completeness than a significant alternative to previously defined parameter variations. Roux[43] recognized the complexity problems in Matula's implementations[40] and mentioned restrictions that could be applied to

overcome them for a polythetic divisive implementation. Unfortunately, no algorithmic details were given.

Overall, the Lance–Williams recurrence formula, and its subsequent extensions, provides a consolidating basis for the implementation of hierarchic agglomerative methods. However, the standard ways of implementation, that is, by generating, storing, and updating the full distance matrix, or by generating distances as required, tend to be very demanding of computational resources. The review by Murtagh[39] explained how substantial reductions in the computational requirements for some of these methods could be achieved by using the reciprocal nearest neighbor approach. El-Hamdouchi and Willett[44] described the use of this approach for the implementation of the Ward method for document clustering. That same year (1989) Rasmussen and Willett[45] discussed parallel implementation of single-link and Ward methods for both document and chemical structure clustering. The RNN approach and single-link clustering have the additional advantage of only requiring a list of descriptor vectors and a function to return the nearest neighbor of any input vector, rather than a full proximity matrix. Downs, Willett, and Fisanick[46] used RNN implementations of the Ward and group-average methods in a comparison of methods for clustering compounds on the basis of property data (see section below on Comparative Studies on Chemical Data Sets). These two agglomerative methods have been used successfully in comparative studies covering a wide range of nonchemical applications, and they have been shown to provide consistently reasonable results. However, centroid- and medoid-based methods, such as Ward (and k-means nonhierarchical), and the group-average and complete-link methods tend to favor similarly sized hyperspherical clusters (i.e., clusters that are shaped like spheres in a space of more than three dimensions), and they can fail to separate clusters of different shapes, densities, or sizes. Single-link is not a centroid method; it uses just the pairwise similarities and is more analogous to density-based methods. Accordingly, it can find clusters of different shapes and sizes, but it is very susceptible to noise, such as outliers, and artifacts, and it has a tendency to produce straggly clusters (an effect known as *chaining*). The development of traditional hierarchical methods largely ignored the issues of noise, and, although the abilities of different methods to separate clusters were noted, little was done about this problem other than to advise users to adopt more than one method so that different types of clusters could be revealed.

Recent developments in the data mining community have produced methods better suited to finding clusters of different shapes, densities, and sizes. For example, Guha, Rastogi, and Shim[47,48] developed an algorithm called ROCK (RObust Clustering using linKs) that is a sort of hybrid between nearest-neighbor, relocation, and hierarchical agglomerative methods. Although more expensive computationally than RNN implementations of the Ward method, the algorithm is particularly well suited to nonnumerical data (of which the Boolean fingerprints used for chemical data sets are a

special case, although they can also be represented as binary, a special case of numeric). The same authors developed another algorithm called CURE (Clustering Using REpresentatives).[49] Here centroid and single-link-type approaches are combined by choosing more than one representative point from each cluster. With CURE, a user-specified number of diverse points is selected from a cluster, so that it is not represented by just the centroid (which tends to lead to hyperspherical clusters). To avoid the problem of influence from selected points that might be outliers, which can result in a chaining effect, the selected points are shrunk toward the cluster centroid by a specified proportion. This results in a computationally more expensive procedure, but the separation of differently shaped and sized clusters is better. Karypis, Han, and Kumar[50] also addressed the problems of cluster shapes and sizes in their Chameleon algorithm. These authors provide a useful overview of the problems of other clustering methods in their summary. Chameleon measures similarity on the basis of a dynamic model, which is to be contrasted with the fixed model of traditional hierarchical methods. Two clusters are merged only if their interconnectivity and closeness is high relative to the internal interconnectivity and closeness within the two clusters. The characteristics of each cluster are thus taken into account during the merging process rather than assuming a fixed model that, if the clusters do not conform to it, can result in inappropriate merging decisions that cannot be undone subsequently. In a different study, Karypis, Han, and Kumar[51] evaluated the use of multilevel refinement methods to detect and correct inappropriate merging decisions in a hierarchy. Fasulo[52] reviewed some of the other recent developments in the area of data mining with World Wide Web search engines. The developments cited in that review describe work that reassesses the manner in which clustering is performed; a range of methods, which are more flexible in their separation of clusters, were evaluated. It is further pointed out that problems still remain when scaling-up hierarchical clustering methods to the very high dimensional spaces characteristic of many chemical data sets. Other fundamental issues remain, such as the problem of tied proximities in hierarchical clustering.[53] This problem was mentioned many years earlier by Jain and Dubes,[28] among others. Tied proximities occur when the proximities between two different pairs of data items are equal, and result in ambiguous decision points when building the hierarchy, effectively leading to many possible hierarchies of which only one is chosen. MacCuish, Nicolaou, and MacCuish[53] show tied proximities to be surprisingly common with the types of fingerprints commonly used in chemical applications, and the problem increases with data set size. What is not clear is whether such ties have a major deleterious effect on the overall clustering and whether the chosen hierarchy is significantly different from any of the others that might have been chosen.

There has been little development of polythetic divisive methods since the publication of the minimum-diameter method[33] in 1991. Garcia et al.[54] developed a path-based approach with similarities to single-link. The method

has time requirements of $O(MN^2)$ for M clusters and N compounds, making the method particularly suitable for finding a small number of clusters. Wang, Yan, and Sriskandarajah[55] updated the single criterion minimum-diameter method with a multiple criteria algorithm that considers both maximum split (intercluster separation) and minimum diameter in deciding the best bipartition. Their algorithm reduces the *dissection* effect (similar items forced into different clusters because doing so reduces the diameter) associated with the minimum-diameter criterion and the chaining effect associated with the maximum-split criterion. More recently, Steinbach, Karypis, and Kumar[56] reported an interesting variant of k-means that is actually a hierarchical polythetic divisive method. At each point where a cluster is to be split into two clusters, the split is determined by using k-means, hence the name "bisecting k-means." The results for document clustering, using keywords as descriptors, are shown to be better than standard k-means, with cluster sizes being more uniform, and better than the agglomerative group-average method.

Monothetic divisive clustering has largely been ignored, although there have been applications and development of a classification method closely related to monothetic divisive clustering. This classification is recursive partitioning, a type of decision tree method.[57–60]

Nonhierarchical algorithms that cluster the data set in a single pass, such as the leader algorithm, have had little development, except to identify appropriate ways of preordering the data set so as to get around the problem of dependency on processing order (work on this is discussed in the Chemical Applications section). For multipass algorithms, however, efforts have been made to minimize the number of passes required, in some cases reducing them to single-pass algorithms. In the area of data mining, this work has resulted in a method that does not fit neatly into the categorization used in this review. Zhang, Ramakrishnan, and Livny[61] developed a program called BIRCH (Balanced Iterative Reducing and Clustering using Heuristics), an $O(N^2)$ method that performs a single scan of the data set to sort items into a cluster features (CF) tree. This operation has some similarity with the leader algorithm; the nodes of the tree store summary information about clusters of dense points in the data so that the original data need not be accessed again during the clustering process. Clustering then proceeds on the in-memory summaries of the data. However, the initial CF tree building requires the maximum cluster diameter to be specified beforehand, and the subsequent tree building is thus sensitive to the value chosen. Overall, the idea of BIRCH is to bring together items that should always be grouped together, with the maximum cluster diameter ensuring that the cluster summaries will all fit into available memory. Ganti et al.[62] outlined a variant of BIRCH called BUBBLE. It does not rely on vector operations but builds up the cluster summaries on the basis of a distance function that obeys the triangle inequality, an operation that is more CPU demanding than operations in coordinate space.

Nearest-neighbor nonhierarchical methods have received much attention in the chemical community because of their fast processing speeds and ease of implementation. The comparative studies outlined in the next section (Comparative Studies on Chemical Data Sets) led to the widespread adoption of the Jarvis–Patrick nearest-neighbor method for clustering large chemical data sets. To improve results obtained by the standard Jarvis–Patrick implementation, several extensions have been developed. The standard implementation tends to produce a few large heterogeneous clusters and an abundance of singletons, which is hardly surprising because the method was originally designed to be space distorting,[34] that is, contraction of sparsely populated areas clusters and splitting of densely populated areas. Attempts to overcome these tendencies include the use of variable-length nearest-neighbor lists,[12,20] reclustering of singletons,[63] and the use of fuzzy clustering.[64] For variable-length nearest-neighbor lists, the user specifies a proximity threshold so that the lists will contain all neighbors that pass the threshold test rather than a fixed number of nearest neighbors. During clustering, the comparison between nearest-neighbor lists is made on the basis of a specified minimum percentage of the neighbors in the shorter list being in common. These modifications help prevent true outliers from being forced to join a cluster while preventing the arbitrary splitting of large clusters arising from the limitations imposed by fixed length lists. When using fingerprints for clustering chemical data sets, Brown and Martin[20] showed improved results compared with the standard implementation, whereas Taraviras, Ivanciuc, and Cabrol-Bass[65] show contrary results when clustering descriptors.

The reclustering of singletons is used in the "cascaded clustering" method of Menard, Lewis, and Mason.[63] This method applies the standard Jarvis–Patrick clustering iteratively, removes the singletons, and reclusters them using less strict parameters until fewer than a specified percentage of singletons remain. The fuzzy Jarvis–Patrick method outlined by Doman et al.[64] is the most radical Jarvis–Patrick variant. In the fuzzy method, clusters in dense regions are extracted using a similarity threshold and the standard crisp method. The compounds are then assigned probabilities of belonging to each of the crisp clusters. Any previously unclustered compounds not exceeding a specified threshold probability of belonging to any of the crisp clusters are regarded as outliers and remain as singletons.

Other nearest-neighbor methods include the agglomerative hierarchical method of Gowda and Krishna,[66] which uses the position of nearest neighbors, rather than just the number, in a measure called the *mutual neighborhood value* (MNV). Given points i and j, if i is the pth neighbor of j and j is the qth neighbor of i, then the MNV is $(p + q)$. Smaller values of MNV indicate greater similarity, and a specified threshold MNV is used to determine whether points should be merged. Dugad and Ahuja[67] extended the MNV concept to include the density of two clusters that are being considered for merger. In addition to the threshold MNV, if there exists a point k with

MNV (*i,k*) less than MNV (*i,j*) but distance (*i,k*) greater than or equal to distance (*i,j*), then *i* is not a valid neighbor of *j*, and *j* is not a valid neighbor of *i*. The neighbor validity check can result in many small clusters, but these clusters can be merged afterward by relaxing the reciprocal nature of the check.

Relocation algorithms are widely used outside of chemical applications, largely because of their simplicity and speed. The original k-means noncombinatorial methods, such as that by Forgy,[68] and the combinatorial methods, such as that by MacQueen,[69] have been modified into different versions for use in many disciplines, a few of which are mentioned here. Efficient implementations of k-means include those by Hartigan and Wong[70] and Spaeth.[71] A variation of the k-means algorithm, referred to as the *moving method*, looks ahead to see whether moving an item from one cluster to another will result in an overall decrease in the square error (Eq. [2]); if it does, then the moving is carried out. Duda and Hart[72] and Ismail and Kamel[73] originally outlined this variant, while Zhang, Wang, and Boyle[74] further developed the idea and obtained better results than a standard noncombinatorial implementation of k-means. Because the method relies on the concept of a centroid, it is usually used with numerical data. However, Huang[75] reported variants that use k-modes and k-prototypes that are suitable for use with categorical and mixed-numerical and categorical data, respectively.

The main problems with k-means are (1) the tendency to find hyperspherical clusters, (2) the danger of falling into local minima, (3) the sensitivity to noise, and (4) the variability in results that depends on the choice of the initial seed points. Because k-means (and its fuzzy equivalent, c-means) is a centroid-based method, nothing much can be done about the tendency to produce hyperspherical clusters, although the CURE methodology mentioned above might alleviate this tendency somewhat. Falling into local minima cannot be avoided, but rerunning k-means with different seeds is a standard way of producing alternative solutions. After a given number of reruns, the solution is chosen that has produced the lowest square-error across the partition. An alternative to this is to perturb an existing solution, rather than starting again. Zhang and Boyle[76] examined the effects of four types of perturbation on the moving method and found little difference between them. Estivell-Castro and Yang[77] suggested that the problem of sensitivity to noise is due to the use of means (and centroids) rather than medians (and medoids). These authors proposed a variant of k-means based on the use of medoids to represent each cluster. However, calculation of a point to represent the medoid is more CPU-expensive [$O(n \log n)$ for each cluster of size n] than that required for the centroid, resulting in a method that is slightly slower than k-means (but faster than EM algorithms[36]). A similar variant based on medoids is the PAM (Partitioning Around Medoids) method developed by Kaufman and Rousseeuw.[2] This method is very time consuming, and so the authors developed CLARA (Clustering LARge Applications), which takes a sample of a data

set and applies PAM to it. An alternative to sampling the compounds has been developed by Ng and Han.[78] Their CLARANS (Clustering Large Applications based on RANdomized Search) method samples the neighbors, rather than the compounds, to make PAM more efficient.

The most common way of choosing seeds for k-means is by random selection, which is statistically reasonable given a large heterogeneous data set. Alternatively, a set of k diverse seeds could be selected using, for example, the MaxMin subset selection method.[79,80] Diverse seeds have been shown to give better clustering results by Fisher, Xu, and Zard.[81] One of the early suggestions, by Milligan,[82] was that a partition resulting from hierarchical agglomerative clustering should be used as the initial partition for k-means to refine. It may seem counterproductive to initialize an $O(N)$ method by first running an $O(N^2)$ method, because it means that very large data sets cannot be processed, but k-means is then effectively being used to refine individual partitions and to correct inappropriate assignments made by the hierarchical method. An iterative method for refining an entire hierarchy has been discussed by Fisher.[83] The iterative method starts at the root (i.e., the top of the hierarchy, with all compounds in one cluster), recursively removes each cluster, resorts it into the hierarchy, and continues iterating until no clusters are moved, other than moving individual items from one cluster to another.

Of the mixture model methods, the expectation maximization (EM) algorithm[36] is the most popular because it is a general and effective method for estimating the model parameters and for fitting the model to the data. Though now quite old, the method was relatively unused until a surge of recent interest has propelled its further development and implementation for data mining applications.[84] As mentioned earlier, k-means is a special case of EM. However, because standard k-means uses the Euclidean metric, it is not appropriate for clustering discrete or categorical data. The EM algorithm does not have these limitations, and, since the mixture model is probabilistic, it can also effectively separate clusters of different sizes, shapes, and densities. A major contribution to the development of the EM algorithm came from Banfield and Raftery[85] who reparameterized the standard distributions to make them more flexible and include a Poisson distribution to account for noise. Various models were developed and compared using the approximate weight of evidence (AWE) statistic, which estimates the Bayesian posterior probability of the clustering solution. Fraley and Raftery[86] subsequently replaced AWE by the more reliable Bayesian information criterion (BIC), which enabled them to produce an EM algorithm that simultaneously yields the best model and determines the best number of clusters. One other interesting aspect of their work is that the EM algorithm is seeded with the clustering results from hierarchical agglomerative clustering. It is not clear whether, by using a less expensive seed selection, the EM algorithm will scale to the very large, high-dimensional data sets of chemical applications, or if the necessary parameterization will be acceptable in practice.

The use of a fixed model in a clustering method favors retrieval of clusters of certain shapes (as exemplified by the hyperspherical clusters retrieved by centroid-based methods). An alternative is to use a density-based approach, in which a cluster is formed from a region of higher density than its surrounding area. The clustering is then based on local criteria, and it can pick out clusters of any shape and internal distribution. Such approaches are typically not applicable directly to high dimensions, but progress is being made in that direction within the data mining community. An example is the DBSCAN (Density-Based Spatial Clustering of Applications with Noise) method of Ester et al.[87] that was subsequently extended by Ankerst et al.[88] to give the OPTICS (Ordering Points To Identify the Clustering Structure) method. These two methods work on a principle that each point of a cluster must have at least a given number of other points within a specified radius. Points fulfilling these conditions are clustered; any remaining points are considered to be outliers, that is, noise. The OPTICS method has been enhanced by Breunig et al.[89] to identify outliers, and by Breunig, Kriegel, and Sander,[90] who combined it with BIRCH[61] to increase speed.

Other density-based approaches designed for high dimensions include CLIQUE (Clustering In QUEst) by Agrawal et al.,[91] and PROCLUS (PROjected CLUSters), by Aggarwal et al.[92] These two methods recognize that high dimensional spaces are typically sparse so that the similarity between two points is determined by a few dimensions, with the other dimensions being irrelevant. Clusters are thus formed by similarity with respect to subspaces rather than full dimensional space. In the CLIQUE algorithm, dense regions of data space are determined by using cell-based partitioning, which are then used as initial bases for forming the clusters. The algorithm works from lower to higher dimensional subspaces by starting from cells identified as dense in $(k-1)$-dimensional subspace and extending them into k-dimensional subspace. The result is a set of overlapping dense regions that are extracted as the clusters. Research into improving grid-based methods is continuing, as demonstrated by the variable grid method of Nagesh.[93] In contrast, the PROCLUS program generates nonoverlapping clusters by identifying potential cluster centers (medoids) using a MaxMin subset selection procedure. The best medoids are selected from the initial set by an iterative procedure in which data items within the locality of a medoid (i.e., within the minimum distance between any two medoids) are assigned to that cluster. Rather than using all dimensions, the dimensions associated with each cluster are used in the Manhattan segmental distance[92] to calculate the distance of an item from the cluster. The Manhattan segmental distance is a normalized form of the Manhattan distance that enables comparison of different clusters with varying numbers of dimensions. (The Manhattan, or city-block, or Hamming, distance is the sum of absolute differences between descriptor values; in contrast, the Euclidean distance is the square root of the sum of squares differences between descriptor values.) Once the best medoids have been

selected, a final single pass over the data set assigns each item to its nearest medoid.

Graph-theoretic algorithms have seen little use in chemical applications. The basis of these methods is some form of a graph in which the vertices are the items in the data set and the edges are the proximities between them. Early methods created clusters by removing edges from a minimum spanning tree or by constructing a Gabriel graph, a relative neighborhood graph, or a Delauney triangulation, but none of these graph-theoretic methods are suitable for high dimensions. Reviews of these methods are given by Jain and Dubes[28] and Matula.[94] Recent advances in computational biology have spurred development of novel graph-theoretic algorithms based on isolating areas called cliques or "almost cliques" (i.e., highly connected subgraphs) from the graph of all pairwise similarities. Examples include the algorithms by Ben-Dor, Shamir, and Yakhini,[95] Hartuv et al.,[96] and Sharan and Shamir[97] that find clusters in gene expression data. Jonyer, Holder, and Cook[98] developed a hierarchical graph-theoretic method that begins with the graph of all pairwise similarities and then iteratively finds subgraphs that maximally compress the graph. The time consumption of these graph-theoretic methods is currently too great to apply to very large data sets.

One way to speed up the clustering process is to implement algorithms on parallel hardware. In the 1980s Murtagh[27,99] outlined a parallel version of the RNN algorithm for hierarchical agglomerative clustering. Also in that decade, Rasmussen, Downs, and Willett[45,100] published research on parallel implementations of Jarvis–Patrick, single-link, and Ward clustering for both document and chemical data sets, and Li and Fang[101] developed parallel algorithms for k-means and single-link clustering. In 1990, Li[102] published a review of parallel algorithms for hierarchical clustering. This in turn elicited a classic riposte from Murtagh[103] to the effect that the parallel algorithms were no better than the more recent $O(N^2)$ serial algorithms. Olson[104] presented $O(N)$ and $O(N \log N)$ algorithms for hierarchical methods using N processors. For chemical applications, in-house parallel implementations include the leader algorithm at the National Cancer Institute[105] and k-means at Eli Lilly[79] (both discussed in the section on Chemical Applications), and commercially available parallel implementations include the highly optimized implementation of Jarvis–Patrick by Daylight[14] and the multiprocessor version of the Ward and group-average methods by Barnard Chemical Information.[12]

Another way of speeding up clustering calculations is to use a quick and rough calculation of distance to assess an initial separation of items and then to apply the more CPU-expensive, full-distance calculation on only those items that were found to cluster using the rough calculation. McCallum, Nigam, and Ungar[106] exploited this idea by using the rough calculation to divide the data into *canopies* (roughly overlapping clusters). Only items within the same canopy, or canopies, were used in the subsequent full-distance calculations to determine nonoverlapping clusters (using, e.g., a hierarchical agglomerative, EM,

or k-means method). The nature of the rough-distance measure used can guarantee that the canopies will be sufficiently broad to encompass all candidates for the ensuing full-distance measure. These ideas to speed up nearest-neighbor searches are similar to the earlier use of bounds on the distance measure, as discussed by Murtagh.[27]

Comparative Studies on Chemical Data Sets

Much of the use of clustering for chemical applications is based on the *similar property principle*.[107] This principle, which holds in many, but certainly not all, structure–property relationships, states that compounds with similar structure are likely to exhibit similar properties. Clustering on the basis of structural descriptors is thus likely to group compounds having similar properties. However, there exist many different clustering methods, each having its own particular characteristics that are likely to affect the composition of the resultant clusters. Consequently, there have been several comparative studies on the performance of different clustering methods when applied to chemical data sets. The first such studies were conducted by Willett and Rubin[5,108–110] in the early 1980s. These studies were highly influential in the subsequent implementation of clustering methods in commercial and in-house software systems used by the pharmaceutical industry. Over 30 hierarchical and nonhierarchical methods were tested on 10 small data sets for which certain properties were known. Clustering was conducted using 2D fingerprints as compound representations. The leave-one-out approach (based on the similar property principle) was used to compare the results of different clustering methods by predicting the property of each compound (as the average of the property of the other members of the cluster) and correlating it with the actual property. High correlations indicate that compounds with similar properties have been clustered together. The results indicated that the Ward hierarchical method gave the best overall performance. But, this method was not well suited to processing large data sets due to the requirement for random access to the fingerprints. The Jarvis–Patrick nonhierarchical method results were almost as good and, because it does not require the fingerprints to be in memory, it became the recommended method.

In the early 1990s, a subsequent study by Downs, Willett, and Fisanick[46] compared the performance of the Ward and group-average agglomerative methods, the minimum-diameter divisive hierarchical method, and the Jarvis–Patrick nonhierarchical method when using dataprints of calculated physicochemical properties. In this assessment, a data set was used that was considerably larger than those used in the original studies.[108–110] The results highlighted the poor performance of the Jarvis–Patrick method in comparison with the hierarchical methods. The hierarchical methods all had similar levels of performance with the minimum-diameter method being slightly better for small numbers of clusters. Brown and Martin[20] then investigated the same

clustering methods to compare their performance for compound selection, using various 2D and 3D fingerprints. Active/inactive data was available for the compounds in the data sets used, so assessment was based on the degree to which clustering separated active from inactive compounds (into nonsingleton clusters). Although the Jarvis–Patrick method was the fastest of the methods, it again gave the poorest results. The results were improved slightly by using a variant of the Jarvis–Patrick method that uses variable rather than fixed-length nearest-neighbor lists.[12] Overall, the Ward method gave the best and most consistent results. The group-average and minimum-diameter methods were broadly similar and only slightly worse in performance than the Ward method.

The influence of the studies summarized above can be seen in the methods subsequently implemented by many other researchers for their applications (see the section on Chemical Applications). One method that was included in the original assessment studies, but not in the later assessments, is k-means. This method did not perform particularly well on the small data sets of the original studies, and the resultant clusters were found to be very dependent on the choice of initial seeds; hence it was not included in the subsequent studies. However, k-means is computationally efficient enough to be of use for very large data sets. Indeed, over the last decade k-means and its variants have been studied extensively and developed for use in other disciplines. Because it is being increasingly used for chemical applications, any future comparisons of clustering methods should include k-means.

How Many Clusters?

A problem associated with the k-means, expectation maximization, and hierarchical methods involves deciding how many ''natural'' (intuitively obvious) clusters exist in a given data set. Determining the number of ''natural'' clusters is one of the most difficult problems in clustering and to date no general solution has been identified. An early contribution from Jain and Dubes[28] discussed the issue of *clustering tendency*, whereby the data set is analyzed first to determine whether it is distributed uniformly. Note that randomly distributed data is not generally uniform, and, because of this, most clustering methods will isolate clusters in random data. To avoid this problem, Lawson and Jurs[111] devised a variation of the Hopkins' statistic that indicates the degree to which a data set contains clusters. McFarland and Gans[112] proposed a method for evaluating the statistical significance of individual clusters by comparing the within-cluster variance with the within-group variance of every other possible subset of the data set with the same number of members. However, for large heterogeneous chemical data sets it can be assumed that the data is not uniformly or randomly distributed, and so the issue becomes one of identifying the most natural clusters.

Nonhierarchical methods such as k-means and EM need to be initialized with k seeds. This presupposes that k is a reasonable estimation of the number

of natural clusters and that the seeds chosen are reasonably close to the centers of these clusters. Epter, Krishnamoorthy, and Zaki[113] published one of the few papers addressing these issues for large data sets. Their solution is applicable to distance-based clustering and involves analysis of the histogram of pairwise distances between data items. For small data sets, all pairwise distances can be used, whereas for large data sets, random sampling (up to 10% of the data set) can be used to lessen the quadratic increase in time needed to generate the distances. For the distances calculated, the corresponding histogram is generated and then scanned to find the first spike (a large maximum followed by a large minimum). This point is used as the threshold for intracluster distance. The graph containing distances within this threshold contains connected components used to determine both the number of clusters present in the data set and the initial starting points from within these clusters. Assuming that a reasonable value for k is known, Fayyad, Reima, and Bradley[114,115] showed that one can minimize the problem of poor initial starting points by sampling the data set to derive a better set of starting points. A series of randomly selected subsets, larger than k, are extracted, clustered by k-means, amalgamated, and then clustered again using each solution from the subsets. The starting points from the subset giving the best clustering of the amalgamated subset are then chosen as the set of refined points for the main clustering, where "best" means the clustering that gives the minimal "distortion," that is, minimum error across the amalgamated subset. The method aims to avoid selecting outliers, which may occur with other selection methods such as MaxMin.

In hierarchical clustering, each level defines a partition of the data set into clusters. However, there is no associated information indicating which level is best in terms of splitting the data set into the "natural" number of clusters present and with each cluster containing the most appropriate compounds. Many methods and criteria have been proposed to try to derive such information from the hierarchy so that the "best" level is selected. Milligan and Cooper[116] published the first comprehensive comparison of hierarchy level selection methods, using psychology data. Thirty methods were tested for their ability to retrieve the correct number of clusters from several small data sets containing from 2 to 5 "natural" clusters. Fifteen years later, Wild and Blankley[117] published a major comparison of hierarchy level selection methods using chemical data sets. As part of that study, Ward clustering with 2D fingerprints was used to evaluate the performance of nine hierarchy level selection methods. The methods chosen were those that would be easy to implement and that did not require parameters. Eight of those methods were ones that Milligan and Cooper had previously examined; the ninth was a more recent method published by Kelley, Gardner, and Sutcliffe.[118] The study by Wild and Blankley concluded that the point biseral,[119] variance ratio criterion,[120] and Kelley methods gave the best overall results, with the Kelley method being more computationally efficient than the others [scaling at less

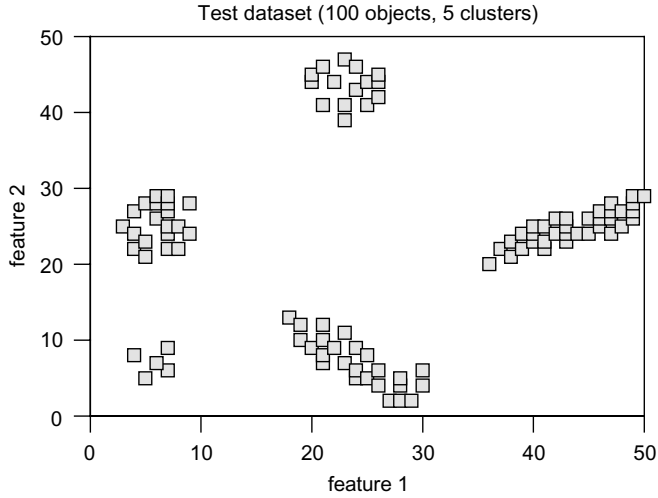

Figure 3 An example data set of 100 objects, represented by 2 features, that fall into 5 natural clusters.

than $O(N^2)$]. A test data set of 100 objects, represented by 2 features and grouped into 5 natural clusters, is shown in Figure 3. The corresponding plot of penalty values (calculated using the Kelley method) against the number of clusters (Figure 4) shows a clear minimum at 5 clusters.

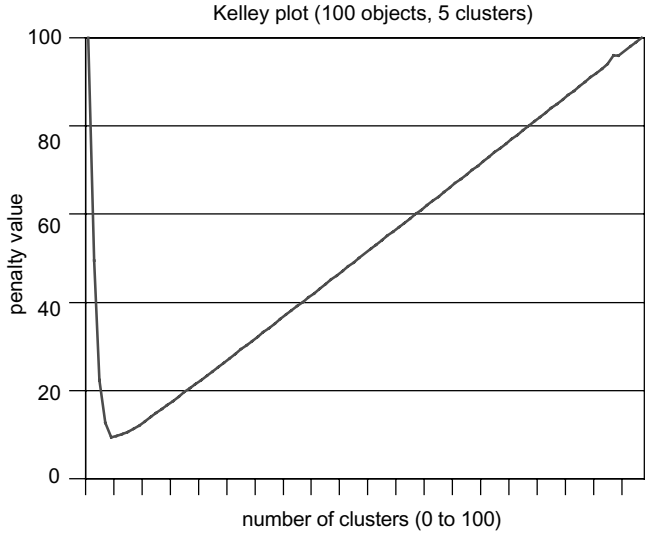

Figure 4 Kelley plot of the penalty value against number of clusters for the data set of 100 items in Figure 3, showing the minimum at 5 clusters.

Hierarchy level selection methods provide useful guidance in selecting reasonable partitions from hierarchies where the underlying structure of the data set is unknown. They are, however, a compromise in that they compare entire partitions with each other rather than individual clusters. In disciplines outside of chemistry, there is an increasing awareness that such global comparisons can mask comparative differences in local densities. For example, the situation in Figure 5 shows three clusters (below the dendrogram) that cannot be retrieved by using a conventional straight horizontal line across the dendrogram (such as that shown in Figure 1). Using a straight line can include either item 8 with cluster [3,1,2] but merge [4,5] with [6,7], or keep [4,5] and [6,7] separate but maintain 8 as a singleton. What may be required for the selection of the "best," nonoverlapping clusters from different partitions is a *stepped* (or *segmental*) horizontal line, which is illustrated by the dotted line across the dendrogram in Figure 5. No solution to deciding which is the best selection of nonoverlapping clusters appears to have been published to date, but there are examples of methods that are moving toward a solution. One such example

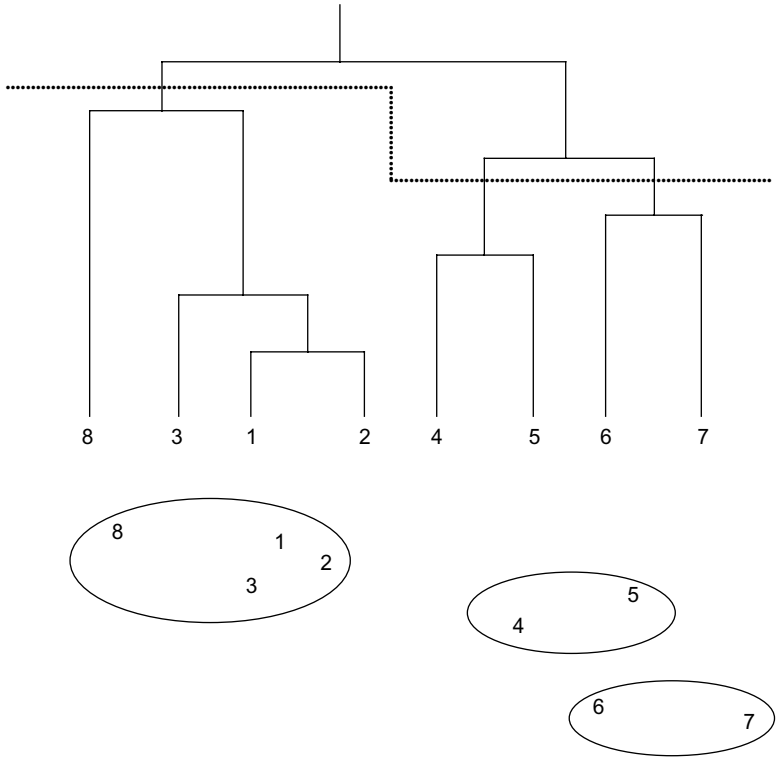

Figure 5 An illustration of how a stepped hierarchy partition can extract particular clusters (clusters [8,1,2,3], [4,5], and [6,7], as shown below the hierarchy).

is the OPTICS method that orders items in a data set in terms of local criteria, thus providing an equivalent to a density-based analysis.

A variety of different requirements exist for chemical applications. These requirements dictate whether it is important to address the issues of how many clusters exist, what the best partition is, and which the best clusters are. When using representative sampling, for example, for high-throughput screening in pharmaceutical research, the number of required clusters is usually set beforehand. Hence, it is necessary to generate only a reasonable partition from which to extract the required number of representative compounds. For analysis of an unknown data set in, say, a list of vendor compounds, the number of clusters is unknown. Hierarchical clustering with optimum level analysis should provide suitable results for this scenario since the actual composition of each cluster is not critical. For analysis of quantitative structure–activity relationships (QSAR), the number of clusters is unknown, and the quality of the clusters becomes an important issue since complete clusters are required for further analysis. It may be that recent developments[87–93] related to density-based clustering will help in this circumstance.

CHEMICAL APPLICATIONS

Having introduced and described the various kinds of clustering methods used in chemistry and other disciplines, we are in a position to present some illustrative examples of chemical applications. This section reviews a representative selection of publications that have reported or analyzed the use of clustering methods for processing chemical data sets, largely from groups of scientists working within pharmaceutical companies. The main applications for these scientists are high-throughput screening, combinatorial chemistry, compound acquisition, and QSAR. The emphasis is on pharmaceutical applications because these workers tend to process very large and high dimensional data sets. This section is presented according to method, starting with hierarchical and then moving to nonhierarchical methods.

Little has been reported on the use of hierarchical divisive methods for processing chemical data sets (other than the inclusion of the minimum-diameter method in some of the comparative studies mentioned above). Recursive partitioning, which is a supervised classification technique very closely related to monothetic divisive clustering, has, however, been used at the GlaxoSmithKline[57,58] and Organon[59] companies.

There is, however, widespread use of hierarchical agglomerative techniques, particularly the Ward method. At Organon, Bayada, Hamersma, and van Geerestein[121] compared Ward clustering with the MaxMin diversity selection method, Kohonen maps, and a simple partitioning method to help select diverse yet representative subsets of compounds for further testing. The data came from HTS or combinatorial library results. Ward clustering

was the only method that gave results consistently better than random selection of compounds. It was also found that the standard technique of selecting the compound closest to the centroid to serve as the representative for a cluster tends to result in the selection of the smallest compound or the one with the fewest features. This finding is not surprising because the centroid is the arithmetic average of items in a cluster and hence the representative will be the most common denominator. Users should be aware of this tendency toward biased selection of a representative compound, since such a compound could be less interesting as a drug-like molecule than others in the data set. This effect was not observed when the clustering was done using the first 10 principal components of the descriptor set rather than relying directly on the descriptors (such as fingerprints) themselves.

Van Geerestein, Hamersma, and van Helden[122] used Ward clustering to show that cluster representatives provide a significantly better sampling of activity space than random selection. This key paper shows how clustering can separate actives from inactives in a data set, so that a cluster containing at least one active will contain more than an average number of other actives. The introduction to their article also gives a succinct summary of why diversity analysis (such as clustering) is of use as a lead finding strategy.

At Parke-Davis (now Pfizer), Wild and Blankley[123] incorporated Ward clustering and level selection (by the Kelley function[118]) into a program called VisualiSAR, which supports structure browsing and the development of structure–activity relationships in HTS data sets. At the Janssen unit of the Johnson and Johnson company, Engels et al.[124] have similarly incorporated Ward clustering and the Kelley function into a system (called CerBeruS) that is used for analysis of their corporate compound database. The clustering was used to produce smaller, more homogeneous subsets from which one representative compound was selected as a screening candidate using the Kelley function to determine the optimal clustering level(s). Engels et al.[124] noted two further advantages of a cluster-based approach. First, if a hit was found, related compounds could be tested subsequently by extracting other possible candidates from the cluster containing the hit, and, second, analyses of structure–activity relationships (SAR) could be formulated by linking the results of all the screening runs so as to examine the cluster hierarchy at different levels. Engels and Venkatarangan[125] subsequently developed a two-stage sequential screening procedure supported by clustering to make HTS more efficient.

Stanton et al.[126] reported the use of complete-link clustering in the HTS system at the Proctor & Gamble company. In situations where the screening produces large numbers of hits, clustering was used to determine which compound classes were present so that representatives could be taken. The amount of follow-up analysis was reduced by an order of magnitude while still evaluating which classes of compounds were present in the hits, thus increasing the efficiency of selecting potential leads. The clusters also provided sets of compounds to build preliminary SAR models. Furthermore, the clustering was

found useful in the detection of false positives, especially from combinatorial libraries. In these cases, the structural similarity between the hits was low and their biological activity was subsequently attributed to a common side product. Clustering was performed by Stanton[127] using BCUT (Burden–CAS–University of Texas) descriptors,[128] with the optimum hierarchy level determined visually from the dendrogram. Visual selection was possible because the hit sets were typically a few hundred compounds.

The most significant application of a nonhierarchical single-pass method was for screening antitumor activity at the National Cancer Institute. A variant of the leader algorithm was developed[129] in which the descriptors were weighted by occurrence in each compound, size of the fragment, and frequency of occurrence in the data set. Because of the use of these weighted descriptors, an asymmetric coefficient[129] was used to determine similarity, rather than the more usual Tanimoto coefficient. The data set was then ordered by the increasing sum of fragment weights to remove the order dependency associated with the leader algorithm (or at least, to have a reasonable basis for choosing a particular order) and to enable the use of heuristics to reduce the number of similarity calculations. Compounds were then assigned to any existing cluster for which they exceeded the given similarity threshold, thus creating overlapping clusters. The algorithm was implemented on parallel hardware,[105] and the results from clustering several data sets were presented with a discussion on the large number of singleton clusters produced.[130] Another variant on the leader algorithm was proposed by Butina.[131] In his approach, the compounds are first sorted by decreasing number of near neighbors (within a specified threshold similarity), thus again removing the order dependence of the basic algorithm. Of course, identifying the number of near neighbors for each compound introduces an $O(N^2)$ step, which in turn obviates the single-pass algorithm's primary advantage of linear speed.

At Rohm and Haas Company, Reynolds, Drucker, and Pfahler[132] developed a two-pass method similar to the initial assignment stage of k-means. In the first pass, a similarity threshold is specified, and then the sphere exclusion diverse subset selection method[80] is used to select the cluster seeds (referred to as *probes*). In the second pass, all other compounds are assigned to the most similar probe (the published version unnecessarily performs this in two stages). Clark and Langton[133] adopted a similar methodology in the Tripos OptiSim fast clustering system for selecting diverse yet representative subsets. OptiSim works by selecting an initial seed at random, selecting a random sample of size K, analyzing the random sample by choosing the most dissimilar member of the sample from existing seeds, and, if the minimum similarity threshold, R, to all existing seeds is exceeded, adding it to the seed set. This operation continues until the specified number of seeds, M, has been selected or no more candidates remain. All other compounds are then assigned to their nearest seed (which is equivalent to the initial assignment stage of k-means, with no refinement). OptiSim is an obvious amalgam of the MaxMin and sphere

exclusion subset selection methods[80] and the Reynolds system mentioned above. It also bears similarities with other methods, particularly the clustering of merged multiple random samples reported by Bradley and Fayyad.[115]

The widespread application of the Jarvis–Patrick nonhierarchical method exists in part because of the influence of the publications by Willett et al.[5,108–110] but also because of the availability of the efficient commercial implementation from Daylight[14] for handling very large data sets. The first publication on the use of Jarvis–Patrick clustering for compound selection from large chemical data sets was from researchers who implemented it at Pfizer Central Research (UK).[134] Clustering was done using 2D fragment descriptors, with calculation of the list of 20 nearest neighbors using the efficient Perry–Willett inverted file approach.[35] After clustering the data set of about 240,000 compounds, singletons were moved to the most similar nonsingleton cluster, and representative compounds were then extracted by generating cluster centroids and selecting the compound closest to each centroid.

Earlier in this chapter, we mentioned the cascaded Jarvis–Patrick[63] and fuzzy Jarvis–Patrick[64] variants. The cascaded Jarvis–Patrick method was implemented at Rhone-Poulenc Rorer (RPR) based on using Daylight 2D structural fingerprints and Daylight's Jarvis–Patrick program. With this variant, singletons are reclustered using less strict parameters so that the singletons do not dominate the set of representative compounds selected. The applications reported by the RPR researchers[63] include selection of compounds from the corporate database for HTS and comparison of the corporate database with external databases, such as the Available Chemicals Directory, to assist in compound acquisition. The fuzzy Jarvis–Patrick variant was developed and implemented at G. D. Searle and Company for analysis of their compound database to help support their screening program. The Searle researchers[64] initially used the Daylight implementation but found the chaining and singleton characteristics of the standard method to be significant drawbacks. This in turn prompted them to develop a variant with different characteristics.

McGregor and Pallai[135] discussed an in-house implementation of the standard Jarvis–Patrick algorithm at Procept Inc. They used the MDL 2D structural descriptors to compare and analyze external databases for efficient compound acquisition. Shemetulskis et al.[136] also reported the use of Jarvis–Patrick clustering to assist in compound acquisition at Parke-Davis, giving results from analysis and comparison of the CAST3D and Maybridge compound databases with the corporate database. In a two-stage process, representatives, comprising about a quarter of the compounds, were selected from each data set by clustering on the basis of 2D fingerprints. Each data set was then merged with the corporate database, and the clustering run again on the basis of calculated physicochemical property descriptors. Clusters containing only CAST3D or Maybridge compounds were tagged as highest priority for acquisition. Dunbar[137] summarized the compound acquisition

report,[135] discussed the use of clustering methods to assist in HTS, and then outlined the use at Parke-Davis of Jarvis–Patrick clustering to assist traditional, low-throughput screening. The aim of the Parke-Davis group was to generate a representative subset of no more than 2000 compounds selected from about 126,000 compounds in the Parke-Davis corporate database so that they could be used in a particularly labor-intensive cell-based assay. Jarvis–Patrick clustering was run to generate an initial set of 25,000 nonsingleton clusters. The compounds closest to the centroids were reclustered to give about 2,300 clusters. The compounds closest to these centroids were then analyzed manually providing a final selection of about 1,400 compounds. An interesting feature of this process was that singletons were rejected at each stage, rather than being assigned to the nearest nonsingleton cluster (as at Pfizer, UK) or being reclustered separately (as in the cascaded clustering method used at Rhone-Poulenc Rorer).

Jarvis–Patrick clustering has also been used to support QSAR analysis in a system developed at the European Communities Joint Research Center.[7,138–140] The EINECS (European Inventory of Existing Chemical Substances) database contains more than 100,000 compounds and has been clustered using 2D structural descriptors. That database also has associated physicochemical properties and activities, but the data is very sparse. Jarvis–Patrick clustering was used to extract clusters containing sufficient compounds with measured data for an attempt to be made to estimate the properties of members of the cluster lacking the data. For a few clusters, it was used to develop reasonable QSAR models.

An example of how use of k-means clustering can be used for QSAR analysis of small data sets is that by Lawson and Jurs[141] who clustered a set of 143 acrylates from the ToSCA (Toxic Substances Control Act) inventory. For large chemical data sets, the seminal paper is that published by Higgs et al.,[79] at Eli Lilly and Company. These authors examined three methods of subset selection to assist their HTS and development of combinatorial libraries. The three methods were k-means, MaxMin, and D-optimal design. Seed compounds were selected by the MaxMin method, and the k-means algorithm was implemented on parallel hardware. This research was part of the compound acquisition strategy to support HTS. The Lilly group used an extensive system of filters to ensure that selected compounds were pharmaceutically acceptable. No recommendations were offered in the paper as to the best method.

The use of a topographic clustering method for chemical data sets is exemplified by the work of Sadowski, Wagener, and Gasteiger.[142] The authors compared three combinatorial libraries using Kohonen mapping. Each compound within a library was represented by a 12-element autocorrelation vector (a sort of 3D-QSAR descriptor). The vectors were used as input to a 50×50 Kohonen network. Mapping the combinatorial libraries onto the same network placed each compound from the library at a particular node in the network. A 2D display of the positions of each compound revealed the degree of

overlap between the libraries. Two very dissimilar libraries formed two distinct clusters with little overlap, whereas two very similar libraries showed no distinction.

The use of mixture-model or density-based clustering has not yet been reported for processing chemical data sets. An interesting application of these techniques is their use to group the compound descriptors so as to obtain a set of orthogonal descriptors. Up to this point, the clustering that we have discussed has been applied to the patterns (fingerprints or dataprints) characterizing each compound; this is the "*Q-mode clustering*" referred to by Sneath and Sokal.[1] One can also cluster the features (the descriptors used in the fingerprints or dataprints) to highlight groups of similar descriptors. Sneath and Sokal call this "*R-mode clustering*." The similar property principle, upon which structure–property relationships depend, assumes that the compound descriptors are independent of each other. Reducing the number of descriptors can thus help in subsequent Q-mode clustering by reducing the dimensionality. Clustering the descriptors, so that a subset of orthogonal descriptors can be extracted, is an alternative to factor analysis and principal components analysis. Using an orthogonal subset of descriptors has the benefit that the result is a set of individual descriptors rather than composite descriptors. Taraviras, Ivanciuc, and Cabrol-Bass[65] applied the single-link, group-average, complete-link, and Ward hierarchical methods, along with Jarvis–Patrick, variable-length Jarvis–Patrick, and k-means nonhierarchical methods to a set of 240 topological indices in an attempt to reveal any "natural" clusters of the descriptors. Descriptors that were found to exist in the same clusters across all seven methods were regarded as being strongly clustered. Reducing the number of methods that needed to be in agreement revealed progressively weaker clusters. Overall, it was found that the strategy of using multiple clustering methods for R-mode clustering could be used to provide representative sets of orthogonal descriptors for use in QSAR analysis.

CONCLUSIONS

Clustering methodology has been developed over many decades. The application of clustering to chemical data sets began in the 1980s, coinciding with the increasing size of in-house compound collections having their information contained in structural databases and with advances made by the information retrieval community to analyze large document collections. In the 1990s the advent of high-throughput screening, combinatorial libraries, and commercially available external chemical inventories placed a greater emphasis on rational compound selection. The demands of clustering data sets of several million compounds with high-dimensional representations led to the widespread adoption of a few inherently efficient and optimally implemented methods, namely, the Jarvis–Patrick, Ward, and k-means methods.

Acceptance of these methods—and inclusion of them as routine operations within such applications as lead-finding strategies, QSAR analyses, and compound acquisition—has been a gradual process rather than an abrupt revolution. The current decade should see this process continue as the methodologies are refined. The push for such advancement appears to be coming again from the information retrieval community but also from the data mining community, which has made significant progress. The emphasis of current research is turning toward the quality of the resultant clusters. It has been shown that, using representatives selected from clusters for lead-finding can increase the active hit rate significantly and consistently.

The results so far in chemistry are promising, but research in other areas outside of chemistry suggests that clustering is still a blunt instrument that can be sharpened by refinements. An example of this refinement is to be able to handle mixed or nonnumerical data, and another example is to take more consideration of cluster sizes, shapes, and distribution. The existing methods and implementations used to analyze chemical data sets do an impressive job when compared with the situation a decade ago. What is exciting is the number of new ideas that are being generated that should result in significant advances in the next decade.

REFERENCES

1. P. H. A. Sneath and R. R. Sokal, *Numerical Taxonomy,* W. H. Freeman, San Francisco, CA, 1973.
2. L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis,* Wiley-Interscience, New York, 1990.
3. B. S. Everitt, *Cluster Analysis,* 3rd ed., Edward Arnold, London, 1993.
4. A. D. Gordon, *Classification,* 2nd ed., Chapman and Hall, London, 1999.
5. P. Willett, *Similarity and Clustering in Chemical Information Systems,* Research Studies Press, Letchworth, UK, 1987.
6. N. Bratchell, *Chemom. Intell. Lab. Systems,* **6**, 105 (1989). Cluster Analysis.
7. J. M. Barnard and G. M. Downs, *J. Chem. Inf. Comput. Sci.,* **32** (6), 644 (1992). Clustering of Chemical Structures on the Basis of Two-Dimensional Similarity Measures.
8. G. M. Downs and P. Willett, in *Advanced Computer-Assisted Techniques in Drug Discovery,* H. van de Waterbeemd, Ed., VCH Publishers, Weinheim, 1994, pp. 111–130. Clustering of Chemical Structure Databases for Compound Selection.
9. R. A. Lewis, S. D. Pickett, and D. E. Clark, in *Reviews in Computational Chemistry,* K. B. Lipkowitz and D. B. Boyd, Eds., Wiley-VCH, New York, 2000, Vol. 16, pp. 1–51. Computer-Aided Molecular Diversity Analysis and Combinatorial Library Design.
10. Clustan Ltd., 16 Kingsburgh Road, Edinburgh, UK. http://www.clustan.com.
11. SAS Institute Inc., SAS Campus Drive, Cary, NC 27513, USA. http://www.sas.com.
12. Barnard Chemical Information Ltd., 46 Uppergate Road, Stannington, Sheffield S6 6BX, UK. http://www.bci.gb.com.
13. Chemical Computing Group Inc., 1010 Sherbrooke Street West, Suite 910, Montreal, Quebec H3A 2R7, Canada. http://www.chemcomp.com.

14. Daylight Chemical Information Systems Inc., 441 Greg Avenue, Santa Fe, NM 87501, USA. http://www.daylight.com.
15. MDL Information Systems, Inc., 14600 Catalina Street, San Leandro, CA 94577, USA. http://www.mdl.com.
16. Accelrys (formerly Molecular Simulations Inc.), 9685 Scranton Road, San Diego, CA 92121-3752, USA. http://www.accelrys.com.
17. Tripos Inc., 1699 South Hanley Road, St. Louis, MO 63144, USA. http://www.tripos.com.
18. W. A. Warr, in *Computational Methods for the Analysis of Molecular Diversity,* P. Willett, Ed., *Perspectives in Drug Discovery and Design,* Vol. 7/8, Kluwer/ESCOM, Dordrecht, The Netherlands, 1997, pp. 115–130. Commercial Software Systems for Diversity Analysis.
19. R. D. Brown, in *Computational Methods for the Analysis of Molecular Diversity,* P. Willett, Ed., *Perspectives in Drug Discovery and Design,* Vol. 7/8, Kluwer/ESCOM, Dordrecht, The Netherlands, 1997, pp. 31–49. Descriptors for Diversity Analysis.
20. R. D. Brown and Y. C. Martin, *J. Chem. Inf. Comput. Sci.,* **36** (*3*), 572 (1996). Use of Structure–Activity Data to Compare Structure-Based Clustering Methods and Descriptors for Use in Compound Selection.
21. R. D. Brown and Y. C. Martin, *J. Chem. Inf. Comput. Sci.,* **37** (*1*), 1 (1997). The Information Content of 2D and 3D Structural Descriptors Relevant to Ligand-Receptor Binding.
22. G. M. Downs and P. Willett, in *Reviews in Computational Chemistry,* K. B. Lipkowitz and D. B. Boyd, Eds., VCH Publishers, New York, 1995, Vol. 7, pp. 1–66. Similarity Searching in Databases of Chemical Structures.
23. P. Willett, J. M. Barnard, and G. M. Downs, *J. Chem. Inf. Comput. Sci.,* **38** (*6*), 983 (1998). Chemical Similarity Searching.
24. J. M. Barnard, G. M. Downs, and P. Willett, in *Virtual Screening for Bioactive Molecules,* H.-J. Böhm and G. Schneider., Eds., Wiley, New York, 2000, pp. 59–80. Descriptor-Based Similarity Measures for Screening Chemical Databases.
25. J. S. Mason and S. D. Pickett, in *Computational Methods for the Analysis of Molecular Diversity,* P. Willett, Ed., *Perspectives in Drug Discovery and Design,* Vol. 7/8, Kluwer/ESCOM, Dordrecht, The Netherlands, 1997, pp. 85–114. Partition-Based Selection.
26. J. A. Hartigan, *Clustering Algorithms,* Wiley, New York, 1975.
27. F. Murtagh, *COMPSTAT Lectures,* **4,** (1985). Multidimensional Clustering Algorithms.
28. A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data,* Prentice Hall, Englewood Cliff, NJ, 1988.
29. G. N. Lance and W. T. Williams, *Computer J.,* **9,** 373 (1967). A General Theory of Classificatory Sorting Strategies. 1. Hierarchical Systems.
30. J. H. Ward, *J. Am. Stat. Assoc.,* **58,** 236 (1963). Hierarchical Grouping to Optimize an Objective Function.
31. E. M. Voorhees, *Inf. Processing Management,* **22** (*6*), 465 (1986). Implementing Agglomerative Hierarchic Clustering Algorithms for Use in Document Retrieval.
32. X. Chen, A. Rusinko III, and S. S. Young, *J. Chem. Inf. Comput. Sci.,* **38** (*6*), 1054 (1998). Recursive Partitioning Analysis of a Large Structure-Activity Data Set Using Three-Dimensional Descriptors.
33. A. Guenoche, P. Hansen, and B. Jaumard, *J. Classification,* **8,** 5 (1991). Efficient Algorithms for Divisive Hierarchical Clustering with the Diameter Criterion.
34. R. A. Jarvis and E. A. Patrick, *IEEE Trans. Computers,* **C-22** (*11*), 1025 (1973). Clustering Using a Similarity Measure Based on Shared Near Neighbors.
35. S. A. Perry and P. Willett, *J. Inf. Sci.,* **6,** 59 (1983). A Review of the Use of Inverted Files for Best Match Searching in Information Rretrieval Systems.
36. A. P. Dempster, N. M. Laird, and D. B. Rubin, *J. Royal Stat. Soc.,* **B39,** 1 (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm.
37. T. Kohonen, *Self-organizing Maps,* Springer-Verlag, Berlin, 1995.

38. J. Zupan and J. Gasteiger, *Neural Networks for Chemists, An Introduction,* VCH, Weinheim, 1993. See also, K. L. Peterson, in *Reviews in Computational Chemistry,* K. B. Lipkowitz and D. B. Boyd, Eds., Wiley-VCH, New York, 2000, Vol. 16, pp. 53–140. Artificial Neural Networks and Their Use in Chemistry.
39. F. Murtagh, in *Handbook of Massive Data Sets,* J. Abello, P. M. Pardalos, and M. G. C. Reisende, Eds., Kluwer, Dordrecht, The Netherlands, 2001, pp. 401–545. Clustering in Massive Data Sets.
40. D. W. Matula, in *Classification as a Tool of Research,* W. Gaul and M. Schader, Eds., Elsevier Science (North-Holland), Amsterdam, 1986, pp. 289–301. Divisive vs. Agglomerative Average Linkage Hierarchical Clustering.
41. N. C. Jain, A. Indrayan, and L. R. Goel, *Pattern Recognition,* **19** (*1*), 95 (1986). Monte Carlo Comparison of Six Hierarchical Clustering Methods on Random Data.
42. J. Podani, *Vegetatio,* **81,** 61 (1989). New Combinatorial Clustering Methods.
43. M. Roux, in *Applied Multivariate Analysis in SAR and Environmental Studies,* J. Devillers and W. Karcher, Eds., Kluwer, Dordrecht, The Netherlands, 1991, pp. 115–135. Basic Procedures in Hierarchical Cluster Analysis.
44. A. El-Hamdouchi and P. Willett, *Computer J.,* **32,** 220 (1989). Hierarchic Document Clustering using Ward's Method.
45. E. M. Rasmussen and P. Willett, *J. Doc.,* **45** (*1*), 1 (1989). Efficiency of Hierarchical Agglomerative Clustering Using the ICL Distributed Array Processor.
46. G. M. Downs, P. Willett, and W. Fisanick, *J. Chem. Inf. Comput. Sci.,* **34** (*5*), 1094 (1994). Similarity Searching and Clustering of Chemical-Structure Databases Using Molecular Property Data.
47. S. Guha, R. Rastogi, and K. Shim, Technical Report, Bell Laboratories, Murray Hill, NJ, 1997. A Clustering Algorithm for Categorical Attributes.
48. S. Guha, R. Rastogi, and K. Shim, *Inf. Systems,* **25** (*5*), 345 (2000). ROCK: A Robust Clustering Algorithm for Categorical Attributes.
49. S. Guha, R. Rastogi, and K. Shim, in *Proceedings of the ACM SIGMOD International Conference on Management of Data,* Seattle, WA, 1998, pp. 73–84. CURE: An Efficient Clustering Algorithm for Large Datasets.
50. G. Karypis, E.-H. Han, and V. Kumar, *IEEE Computer: Special Issue on Data Analysis and Mining,* **32** (*8*), 68 (1999). Chameleon: A Hierarchical Clustering Algorithm Using Dynamic Modeling.
51. G. Karypis, E.-H. Han, and V. Kumar, *Technical Report No. 99-020,* Department of Computer Science & Engineering, University of Minnesota, Minneapolis, MN, 1999. Multilevel Refinement for Hierarchical Clustering.
52. D. Fasulo, *Technical Report No. 01-03-02,* Department of Computer Science & Engineering, University of Washington, Seattle, WA, 1999. An Analysis of Recent Work on Clustering Algorithms.
53. J. MacCuish, C. Nicolaou, and N. E. MacCuish, *J. Chem. Inf. Comput. Sci.,* **41** (*1*), 134 (2001). Ties in Proximity and Clustering Compounds.
54. J. A. Garcia, J. Fdez-Valdivia, J. F. Cortijo, and R. Molina, *Signal Processing,* **44** (*2*), 181 (1994). A Dynamic Approach for Clustering Data.
55. Y. Wang, H. Yan, and C. Sriskandarajah, *J. Classification,* **13,** 231 (1996). The Weighted Sum of Split and Diameter Clustering.
56. M. Steinbach, G. Karypis, and V. Kumar, *Technical Report 00-034,* Department of Computer Science & Engineering, University of Minnesota, Minneapolis, MN, 2000. A Comparison of Document Clustering Techniques.
57. D. M. Hawkins, S. S. Young, and A. Rusinko, *Quant. Struct.–Act. Relat.,* **16,** 396 (1997). Analysis of a Large Structure–Activity Data Set Using Recursive Partitioning.
58. X. Chen, A. Rusinko, and S. S. Young, *J. Chem. Inf. Comput. Sci.,* **38** (*6*), 1054 (1998). Recursive Partitioning Analysis of a Large Structure–Activity Data Set Using Three-Dimensional Descriptors.

59. M. Wagener and V. J. van Geerestein, *J. Chem. Inf. Comput. Sci.,* **40** (*2*), 280 (2000). Potential Drugs and Nondrugs: Prediction and Identification of Important Structural Features.

60. D. W. Miller, *J. Chem. Inf. Comput. Sci.,* **41** (*1*), 168 (2001). Results of a New Classification Algorithm Combining K Nearest Neighbors and Recursive Partitioning.

61. T. Zhang, R. Ramakrishnan, and M. Livny, *ACM SIGMOD Record,* **25** (*2*), 103 (1996). BIRCH: An Efficient Data Clustering Method for Very Large Databases.

62. V. Ganti, R. Ramakrishnan, J. Gehrke, A. Powell, and J. French, *Proceedings of the 15th International Conference on Data Engineering,* Sydney, Australia, 1999, pp. 502–511. Clustering Large Datasets in Arbitrary Metric Spaces.

63. P. R. Menard, R. A. Lewis, and J. S. Mason, *J. Chem. Inf. Comput. Sci.,* **38** (*3*), 379 (1998). Rational Screening Set Design and Compound Selection: Cascaded Clustering.

64. T. N. Doman, J. M. Cibulskis, M. J. Cibulskis, P. D. McCray, and D. P. Spangler, *J. Chem. Inf. Comput. Sci.,* **36** (*6*), 1195 (1996). Algorithm5: A Technique for Fuzzy Clustering of Chemical Inventories.

65. S. L. Taraviras, O. Ivanciuc, and D. Cabrol-Bass, *J. Chem. Inf. Comput. Sci.,* **40** (*5*), 1128 (2000). Identification of Groupings of Graph Theoretical Descriptors Using a Hybrid Cluster Analysis Approach.

66. K. C. Gowda and G. Krishna, *Pattern Recognition,* **10** (*2*), 105 (1978). Agglomerative Clustering Using the Concept of Mutual Nearest Neighborhood.

67. R. Dugad and N. Ahuja, *IEEE International Conference on Acoustics Speech and Signal Processing,* **5,** 2761 (1998). Unsupervised Multidimensional Hierarchical Clustering.

68. E. Forgy, *Biometrics,* **21,** 768 (1965). Cluster Analysis of Multivariate Data: Efficiency vs. Interpretability of Classifications.

69. J. MacQueen, in *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability,* University of California Press, Berkeley, CA, 1967, Vol. 1, pp. 281–297. Some Methods for Classification and Analysis of Multivariate Observations.

70. J. A. Hartigan and M. A. Wong, *Appl. Stat.,* **28,** 100 (1979). A K-Means Clustering Algorithm.

71. H. Spaeth, *Eur. J. Operat. Res.,* **1,** 23 (1977). Computational Experiences with the Exchange Method.

72. R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis,* Wiley, New York, 1973.

73. M. A. Ismail and M. S. Kamel, *Pattern Recognition,* **22,** 75 (1989). Multidimensional Data Clustering Utilizing Hybrid Search Strategies.

74. Q. Zhang, Q. R. Wang, and R. D. Boyle, *Pattern Recognition,* **24** (*4*), 331 (1991). A Clustering Algorithm for Datasets with a Large Number of Classes.

75. Z. Huang, *Int. J. Data Mining Knowledge Disc.,* **2** (*3*), 283 (1998). Extensions to the K-Means Algorithm for Clustering Large Data Sets with Categorical Values.

76. Q. Zhang and R. D. Boyle, *Pattern Recognition,* **24** (*9*), 835 (1991). A New Clustering Algorithm with Multiple Runs of Iterative Procedures.

77. V. Estivell-Castro and J. Yang, *Technical Report No. 99-03,* Department of Computer Science & Software Engineering, University of Newcastle, Callaghan, NSW 2308, Australia. A Fast and Robust General Purpose Clustering Algorithm.

78. R. T. Ng and J. Han, in *Proceedings of the 20th International Conference on Very Large Data Bases,* Santiago, Chile, 1994, pp. 144–155. Efficient and Effective Clustering Methods for Spatial Data Mining.

79. R. E. Higgs, K. G. Bemis, I. A. Watson, and J. H. Wikel, *J. Chem. Inf. Comput. Sci.,* **37** (*5*), 861 (1997). Experimental Designs for Selecting Molecules from Large Chemical Databases.

80. M. Snarey, N. K. Terrett, P. Willett, and D. J. Wilton, *J. Mol. Graphics Modell.,* **15,** 372 (1997). Comparison of Algorithms for Dissimilarity-Based Compound Selection.

81. D. Fisher, L. Xu, and N. Zard, in *Proceedings of the 6th International Workshop on Machine Learning,* Morgan Kaufmann, Aberdeen, UK, 1992, pp. 163–168. Ordering Effects in Clustering.

82. G. W. Milligan, *Psychometrika,* **45** (*3*), 325 (1980). An Examination of the Effect of Six Types of Error Perturbation on Fifteen Clustering Algorithms.

83. D. Fisher, *J. Artif. Intell. Res.,* **4,** 147 (1996). Iterative Optimization and Simplification of Hierarchical Clusterings.

84. G. J. McLachlan and T. Krishnan, *The EM Algorithm and Extensions,* Wiley, New York, 1997.

85. J. D. Banfield and A. E. Raftery, *Biometrics,* **49,** 803 (1993). Model-Based Gaussian and Non-Gaussian Clustering.

86. C. Fraley and A. E. Raftery, *Computer J.,* **41** (*8*), 578 (1988). How Many Clusters? Which Clustering Method? Answers via Model-Based Cluster Analysis.

87. M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, in *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining,* Portland, OR, 1996, pp. 226–231. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise.

88. M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, in *Proceedings of the ACM SIGMOD International Conference on Management of Data,* Philadelphia, PA, 1999, pp. 49–60. OPTICS: Ordering Points to Identify Clustering Structure.

89. M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, *Proceedings of the Conference on Data Mining and Knowledge Discovery,* Prague, Czech Repub., 1999, in *Lecture Notes in Computer* Science, Springer, **1704,** 262–270 (1999). OPTICS-OF: Identifying Local Outliers.

90. M. M. Breunig, H.-P. Kriegel, and J. Sander, in *Proceedings of the European Conference on Principles and Practice of Knowledge Discovery in Databases,* Lyon, France, 2000. Fast Hierarchical Clustering Based on Compressed Data and OPTICS.

91. R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan, in *Proceedings of the ACM SIGMOD International Conference on Management of Data,* Seattle, WA, 1998, pp. 94–105. Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications.

92. C. C. Aggarwal, C. Procopiuc, J. L. Wolf, P. S. Yu, and J. S. Park, in *Proceedings ACM SIGMOD International Conference on Management of Data,* Philadelphia, PA, 1999, pp. 61–72. Fast Algorithms for Projected Clustering.

93. H. S. Nagesh, M.Sc. Thesis, Northwestern University of Illinois, Evanston, IL, 1999. High Performance Subspace Clustering for Massive Data Sets.

94. D. W. Matula, in *Classification and Clustering,* J. van Ryzin, Ed., Academic Press, 1977, pp. 95–129. Graph Theoretic Techniques for Cluster Analysis Algorithms.

95. A. Ben-Dor, R. Shamir, and Z. Yakhini, *J. Comput. Biol.,* **6** (*3/4*), 281 (1999). Clustering Gene Expression Patterns.

96. E. Hartuv, A. Schmitt, J. Lange, S. Meier-Ewart, H. Lehrach, and R. Shamir, in *Proceedings 3rd International Conference on Computational Molecular Biology (RECOMB 99),* Lyon, France, 1999. An Algorithm for Clustering cDNAs for Gene Expression Analysis.

97. R. Sharan and R. Shamir, in *Proceedings of the 8th International Conference on Intelligent Systems for Molecular Biology,* AAAI Press, Menlo Park, CA, 2000, pp. 307–316. CLICK: A Clustering Algorithm with Application to Gene Expression Analysis.

98. I. Jonyer, L. B. Holder, and D. J. Cook, in *Proceedings of the 13th Annual Florida AI Research Symposium,* pp. 91–95, 2000 (http://www-cse.uta.edu/~cook/pubs). Graph-Based Hierarchical Conceptual Clustering.

99. F. Murtagh, *Computer J.,* **26** (*4*), 354 (1983). A Survey of Recent Advances in Hierarchical Clustering Algorithms.

100. E. M. Rasmussen, G. M. Downs, and P. Willett, *J. Comput. Chem.,* **9** (*4*), 378 (1988). Automatic Classification of Chemical Structure Databases Using a Highly Parallel Array Processor.

101. X. Li and Z. Fang, *Parallel Computing,* **11,** 275 (1989). Parallel Clustering Algorithms.

102. X. Li, *IEEE Trans. Pattern Anal. Machine Intelligence,* **12** (*11*), 1088 (1990). Parallel Algorithms for Hierarchical Clustering and Cluster Validity.

103. F. Murtagh, *IEEE Trans. Pattern Anal. Machine Intelligence,* **14** (*10*), 1056 (1992). Comments on "Parallel Algorithms for Hierarchical Clustering and Cluster Validity".

104. C. F. Olson, *Technical Report CSD-94-786,* University of California, Berkeley, CA, 1994. Parallel Algorithms for Hierarchical Clustering.

105. R. Whaley and L. Hodes, *J. Chem. Inf. Comput. Sci.,* **31** (*2*), 345 (1991). Clustering a Large Number of Compounds. 2. Using a Connection Machine.

106. A. McCallum, K. Nigam, and L. H. Ungar, in *Advances in Knowledge Discovery and Data Mining,* MIT Press, Cambridge, MA, 2000. Efficient Clustering of High Dimensional Data Sets with Application to Reference Matching.

107. M. A. Johnson and G. M. Maggiora, *Concepts and Applications of Molecular Similarity,* Wiley, New York, 1990.

108. P. Willett, *Anal. Chim. Acta,* **136,** 29 (1982). A Comparison of Some Hierarchical Agglomerative Clustering Algorithms for Structure–Property Correlation.

109. V. Rubin and P. Willett, *Anal. Chim. Acta,* **151,** 161 (1983). A Comparison of Some Hierarchical Monothetic Divisive Clustering Algorithms for Structure–Property Correlation.

110. P. Willett, *J. Chem. Inf. Comput. Sci.,* **24** (*1*), 29 (1984). Evaluation of Relocation Clustering Algorithms for the Automatic Classification of Chemical Structures.

111. R. G. Lawson and P. C. Jurs, *J. Chem. Inf. Comput. Sci.,* **30** (*1*), 36 (1990). New Index for Clustering Tendency and Its Application to Chemical Problems.

112. J. W. McFarland and D. J. Gans, *J. Med. Chem.,* **29,** 505–514 (1986). On the Significance of Clusters in Graphical Display of Structure–Activity Data.

113. S. Epter, M. Krishnamoorthy, and M. Zaki, *Technical Report No. 99-6,* Computer Science Department, Rensselaer Polytechnic Institute, Troy, NY, 1999. Clusterability Detection and Initial Seed Selection in Large Data Sets.

114. U. M. Fayyad, C. A. Reima, and P. S. Bradley, in *Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining,* R. Agrawal, P. Stolorz, and G. Piatetsky-Shapiro, Eds., AAAI Press, Menlo Park, CA, 1998, pp. 194–198. Initialization of Iterative Refinement Clustering Algorithms.

115. P. S. Bradley and U. M. Fayyad, in *Proceedings of the 15th International Conference on Machine Learning,* Morgan Kaufmann, San Francisco, 1998, pp. 91–99. Refining Initial Points for K-Means Clustering.

116. G. W. Milligan and M. C. Cooper, *Psychometrika,* **50** (*2*), 159 (1985). An Examination of Procedures for Determining the Number of Clusters in a Data Set.

117. D. J. Wild and C. J. Blankley, *J. Chem. Inf. Comput. Sci.,* **40** (*1*), 155 (2000). Comparison of 2D Fingerprint Types and Hierarchy Level Selection Methods for Structural Grouping Using Ward's Clustering.

118. L. A. Kelley, S. P. Gardner, and M. J. Sutcliffe, *Protein Eng.,* **9,** 1063 (1996). An Automated Approach for Clustering an Ensemble of NMR-Derived Protein Structures into Conformationally-Related Subfamilies.

119. G. W. Milligan, *Psychometrika,* **46** (*2*), 187 (1981). A Monte Carlo Study of Thirty Internal Criterion Measures for Cluster Analysis.

120. T. Calinski and J. Harabasz, *Commun. Stat.,* **3** (*1*), 1 (1974). A Dendrite Method for Cluster Analysis.

121. D. M. Bayada, H. Hamersma, and V. J. van Geerestein, *J. Chem. Inf. Comput. Sci.,* **39** (*1*), 1 (1999). Molecular Diversity and Representativity in Chemical Databases.

122. V. J. van Geerestein, H. Hamersma, and S. P. van Helden, in *Computer-Assisted Lead Finding and Optimization* (Proceedings 9th European QSAR Meeting, Lausanne, Switzerland, 1996), H. van de Waterbeemd, B. Testa, and G. Folkers, Eds., Wiley-VCH, Basel, Switzerland, 1997, pp. 159–178. Exploiting Molecular Diversity: Pharmacophore Searching and Compound Clustering.

123. D. J. Wild and C. J. Blankley, *J. Mol. Graphics Modell.,* **17** (*2*), 85 (1999). VisualiSAR: A Web-Based Application for Clustering, Structure Browsing, and Structure–Activity Relationship Study.

124. M. F. M. Engels, T. Thielmans, D. Verbinnen, J. P. Tollenaere,, and R. Verbeeck, *J. Chem. Inf. Comput. Sci.,* **40** (*2*), 241 (2000). CerBeruS: A System Supporting the Sequential Screening Process.

125. M. F. M. Engels and P. Venkatarangan, *Curr. Opin. Drug Discovery Dev.,* **4** (*3*), 275 (2001). Smart Screening: Approaches to Efficient HTS.

126. D. T. Stanton, T. W. Morris, S. Roychoudhury, and C. N. Parker, *J. Chem. Inf. Comput. Sci.,* **39** (*1*), 21 (1999). Application of Nearest-Neighbor and Cluster Analyses in Pharmaceutical Lead Discovery.

127. D. T. Stanton, *J. Chem. Inf. Comput. Sci.,* **39** (*1*), 11 (1999). Evaluation and Use of BCUT Descriptors in QSAR and QSPR Studies.

128. R. S. Pearlman and K. M. Smith, *J. Chem. Inf. Comput. Sci.,* **39** (*1*) 28–35 (1999). Metric Validation and the Receptor-Relevant Subspace Concept.

129. L. Hodes, *J. Chem. Inf. Comput. Sci.,* **29** (*2*), 66 (1989). Clustering a Large Number of Compounds. 1. Establishing the Method on an Initial Sample.

130. R. Whaley and L. Hodes, *J. Chem. Inf. Comput. Sci.,* **31** (*2*), 347 (1991). Clustering a Large Number of Compounds. 3. The Limits of Classification.

131. D. Butina, *J. Chem. Inf. Comput. Sci.,* **39** (*4*), 747 (1999). Unsupervised Data Base Clustering Based on Daylight's Fingerprint and Tanimoto Similarity a Fast and Automated Way to Cluster Small and Large Data Sets.

132. C. H. Reynolds, R. Druker, and L. B. Pfahler, *J. Chem. Inf. Comput. Sci.,* **38** (*2*), 305 (1998). Lead Discovery Using Stochastic Cluster Analysis (SCA): A New Method for Clustering Structurally Similar Compounds.

133. R. D. Clark and W. J. Langton, *J. Chem. Inf. Comput. Sci.,* **38** (*6*), 1079 (1998). Balancing Representativeness Against Diversity Using Optimizable K-dissimilarity and Hierarchical Clustering.

134. P. Willett, V. Winterman, and D. Bawden, *J. Chem. Inf. Comput. Sci.,* **26** (*3*), 109 (1986). Implementation of Nonhierarchic Cluster-Analysis Methods in Chemical Information Systems; Selection of Compounds for Biological Testing and Clustering of Substructure Search Output.

135. M. J. McGregor and P. V. Pallai, *J. Chem. Inf. Comput. Sci.,* **37** (*3*), 443 (1997). Clustering Large Databases of Compounds: Using the MDL "Keys" as Structural Descriptors.

136. N. E. Shemetulskis, J. B. Dunbar, B. W. Dunbar, D. W. Moreland, and C. Humblet, *J. Comput.-Aided Mol. Design,* **9,** 407 (1995). Enhancing the Diversity of a Corporate Database using Chemical Database Clustering and Analysis.

137. J. B. Dunbar, in *Computational Methods for the Analysis of Molecular Diversity,* P. Willett, Ed., *Perspectives in Drug Discovery and Design,* Vol. 7/8, Kluwer/ESCOM, Dordrecht, The Netherlands, 1997, pp. 51–63. Cluster-Based Selection.

138. G. M. Downs and P. Willett, in *Applied Multivariate Analysis in SAR and Environmental Studies,* J. Devillers and W. Karcher, Eds., Kluwer, Dordrecht, The Netherlands, 1991, pp. 247–279. The Use of Similarity and Clustering Techniques for the Prediction of Molecular Properties.

139. J. Nouwen and B. Hansen, *SAR and QSAR in Environmental Research,* **4,** 1 (1995). An Investigation of Clustering as a Tool in Quantitative Structure–Activity Relationships (QSARs).

140. J. Nouwen, F. Lindgren, B. Hansen, W. Karcher, H. J. M. Verhaar, and J. L. M. Hermens, *J. Chemometrics,* **10,** 385 (1996). Fast Screening of Large Databases Using Clustering and PCA based on Structure Fragments.

141. R. G. Lawson and P. C. Jurs, *J. Chem. Inf. Comput. Sci.,* **30** (*2*), 137 (1990). Cluster Analysis of Acrylates to Guide Sampling for Toxicity Testing.

142. J. Sadowski, M. Wagener, and J. Gasteiger, *Angew. Chem., Int. Ed. Engl.,* **34** (*23/24*), 2674 (1995/1996). Assessing Similarity and Diversity of Combinatorial Libraries by Spatial Autocorrelation Functions and Neural Networks.