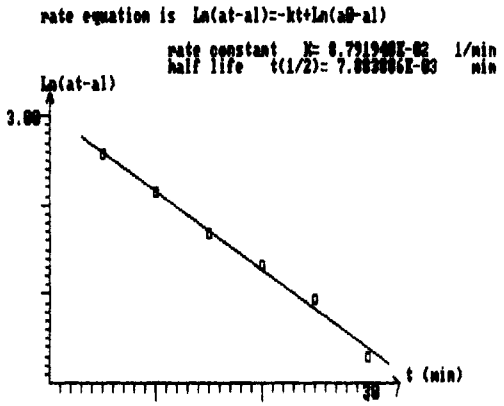


Figure 11. Operational results of Figure 10.

selves as succinct and transparent in logic, easy to read, remember, and understand. After the PAD structure is keyed into the computer, the operational results are shown in Figure 11.

CONCLUSION

From the above examples and the comparison of the program structure described between PAD and FC (see Table I) as well as the comparison between Figure 8 (the universal PAD graphic for calculating the pH of various weak acid balanced systems) and Figure 9 (the FC graphic for calculating the pH of various weak acid balanced systems), we obtain the following conclusions:

(1) Demonstrating the program by using PAD makes the program structure simplified, the logic succinct and transparent.

(2) It is easy and convenient to converse PAD into a source program (compiled manually or automatically by the computer).

(3) By use of PAD in structured programming, the program is easy to read, remember, and understand.

With the support of the PAD system, the computer can do many things such as compile programs, operate programs, and output operational results automatically according to PAD keyed in by the users. Thus, the efficiency of programming, editing, manufacturing, and checking can be raised by reducing many problems met with by workers who are not computer professionals. For this reason, to apply PAD in dealing with various problems in chemistry is a step forward in applying software engineering to the field of chemistry as well as a leap in the history of computing chemistry.

REFERENCES

(1) Nimura, Yoshihiko. In *Program Technique—PAD Structured Programming*; OMU Co.: Japan, 1985.
(2) Guilan, Yan; Jiayao, Liu. Lecture on Software Engineering. *Computer Studies in Fujian*; Fujian Publishing House: Fujian, China, 1986; pp 1-4.
(3) Kennedy, J. H. *J. Chem. Educ.* **1982**, *59*, 523.
(4) Blakely, G. R. *J. Chem. Educ.* **1982**, *59*, 728.
(5) Xianmin, Zheng. pH Value Calculation for Weak Acid (Base) Balanced System—PAD Programming Technique. *Proceedings of 2nd Computing Chemistry of the Chinese Chemical Society*; Jiangxi Publishing House: Jiangxi, China, 1988; Vol. 7, p 26.
(6) Peters, Lawrence J.; Belady, L. A. *Software Design: Methods & Techniques*; Yourdon: Englewood Cliffs, NJ, 1981.

Clustering a Large Number of Compounds. 1. Establishing the Method on an Initial Sample

LOUIS HODES

National Cancer Institute, Bethesda, Maryland 20892

Received March 25, 1988

The National Cancer Institute Division of Cancer Treatment has revised its drug-screening program. About 230 000 compounds in our repository are available for screening under the new protocol. This paper is the first on an attempt to extract a representative sample of these compounds by clustering. It reviews the establishment of the clustering method on a 4980-compound initial sample. The clustering algorithm is fairly simple. However, the molecular fragments employed to match the compounds are somewhat complex to distinguish a large number of compounds.

INTRODUCTION

The National Cancer Institute (NCI) Division of Cancer Treatment (DCT) Developmental Therapeutics Program (DTP) has been converting its primary screening program from in vivo mouse models to cell cultures derived from human cancers.[1] It should soon be possible to screen a large number of compounds, of which many are available from our store of several hundred thousand compounds that have been acquired over the years of our program for the earlier screens.

A search of our file revealed 232 000 compounds with inventory sufficient for this new test. The work reported here is an effort to find a representative sample of these compounds for large-scale testing on the new screens.

Such a sample may be obtained by clustering the compounds according to molecular structure. One or more compounds can be chosen from each cluster.

We assume that compounds with similar structure tend to have similar test results. However, this is only a first approximation, and there are many counterexamples. Therefore, we require the compounds in a cluster to be very much alike. Ideally, they should differ in only one functional group. Such a strict criterion will yield a relatively large number of clusters, which agrees with our need to test as many different substances as possible.

There is also the question as to whether this job stretches the limits of current computer capability. That is, the sheer amount of data may render the project infeasible.

In this paper we present work on an initial sample of 4980 compounds to show how the clustering was developed to yield a satisfactory separation of compounds. The sequel will describe the use of a new species of computer to accomplish the large clustering.

Much work on clustering of chemical structures has been done by Willett.[2] Willett et al.[3] have a discussion on the use of clustering to select compounds for biological testing. Since he has experimented with and reviewed a variety of methods,

0095-2338/89/1629-0066$01.50/0

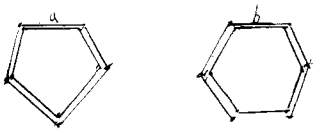


Figure 1. Plain bond centered fragments, centered on bonds a and b. Identical end atoms in fragment a allow discrimination between five- and six-membered rings.

we have generally used Willett's work to compare and confirm our more specialized approach. Throughout this paper we will notice similarities to and differences from his work.

In the next section we discuss the criteria for matching two compounds. This involves developing molecular fragments to be used for matching, their weighting, and the similarity measure for comparing compounds. The section after that will present details on the algorithm used to form the clusters. Experiments performed to optimize weighting factors are then reviewed. Finally, we will examine some preliminary results.

MATCHING ON MOLECULAR FRAGMENTS

Limitations on our data and computation time restrict us to topological features rather than more realistic three-dimensional structure. Similarly, finding a maximal common substructure for each pair of compounds would be computationally extravagant. Also, it makes more sense to use connected fragments to approximate locally connected functional groups, rather than to try for global features. This has the advantage of keeping the number of features for each molecule about the same as its size, i.e., the number of atoms.

Since we are matching compounds by structure, we want the fragments extracted from each compound to be a reasonable approximation of all its chemical features. Ideally, every distinctive feature of a molecule should be represented as a fragment, with a weight approximating its relevance. Then we can match molecules by using a weighted sum of the fragments they have in common.

We view the fragments in detail since the clustering is ultimately limited by the fragment features. First we describe the choice of fragments, then the weighting scheme, and finally the similarity measure.

(A) Fragment Generation. The first choice is whether to use a fixed set of fragments, some standard dictionary as in Willett's work, or a scheme for exhaustive generation. Since we use fairly large fragments with branching, we require exhaustive generation so that we do not miss some unforeseen functional group.

Since features vary in complexity, we generate fragments in three size categories. The smallest consists of single-atom fragments for the usually infrequent but often highly significant metallic elements. Any atom that is not C, O, N, S, H, Cl, P, Br, F, Si, I, As, Se, B, or Te is taken as a single-atom fragment with its numerical metal code. A generic code translating to the letter M is then substituted for the metal atom in the structure for further processing to obtain the larger fragments. Thus, compounds with different metals in the same configuration yield matching fragments at the larger sizes.

The remaining two sizes are both bond-centered. Bonds are distinguished as ring or nonring as well as single, double, triple, tautomer, and, if ring, alternating. Each bond in the molecule determines a fragment, its size depending on the nature of the bond and its attached atoms. We call a bond plain if it is C–C single or C–C ring alternating. Otherwise, it is called special.

The special bonds are usually more interesting than the plain bonds and therefore are restricted to generate smaller fragments. These encompass an additional bond and atom from both original bond atoms. All branching is included.

A plain bond can also generate an extra fragment at this stage if it terminates with special bonds at both ends. Here, only the plain bond and all adjoining special bonds are included. These fragments are grouped with the medium-size fragments generated by special bonds.

Table I. Summary of Fragment Types

type	size	generation
single atom metal	small	all atoms not C, O, N, S, ..., Te
special bond I	medium	all special bonds with all branching one more level
special bond II	medium	all plain bonds terminating at both ends with special bonds and just those bonds
plain bond centered	large	all plain bonds with all branching, two more levels

Table II. Multiplicity Weight for Six Carbon Chain Fragment

multiplicity	wt	multiplicity	wt
1	1	16–31	5
2–3	2	32–63	6
4–7	3	64–127	7
8–15	4		

Generally, the plain bonds extend two levels from both bond atoms, so they can include significant portions of even large molecules. In the case of a simple ring or chain without branching, each fragment centered on a plain bond will include six atoms, provided that the original bond is not near the end of a chain (see Figure 1). The atoms of a fragment may repeat when there are ring structures. Therefore, fragments originated by plain bonds can identify rings of sizes up to five atoms and thus can distinguish five- from six-membered rings. This would not hold for atom-centered fragments extending also two atoms in all directions, i.e., the "frels" used in the DARC[4] system. These would identify rings of sizes up to four atoms only.

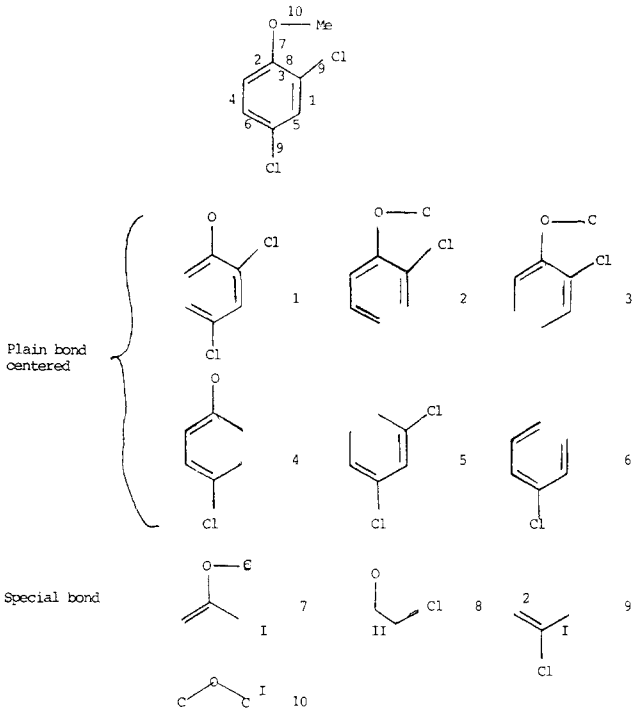
Table I reviews the types of fragments. Note that each compound yields as many fragments as it has bonds plus any metal atoms plus any plain bonds situated between special bonds. This count includes identical fragments; the average number of distinct fragments in our filed compounds is about 12. The 4980 compounds yielded about 9000 distinct fragments, of which about 5000 occurred in a single compound.

Figure 2 illustrates the fragmentation of a typical compound. The fragments are large with an accompanying high degree of specificity when compared, for example, with those used by Willett. This specificity is necessary for greater discrimination on the large set of compounds. In a study[5] of fragments used for structure–activity work, it was found that larger fragments improved performance when the number of compounds increased.

(B) Fragment Weights. The fragments are assigned weights according to their multiplicity in the molecule, their size, and their frequency in the file. The three component weights are combined to form a composite fragment weight. Fragment weights were determined empirically, during many runs, in attempts to improve the clustering, as will be described later.

(1) Multiplicity Weights. A count of the number of occurrences of each fragment in the molecule is produced by the fragment-generating program. As was just mentioned, this count is combined into the weight attached to the fragment for that molecule. Thus, when two molecules have the same fragment, the contribution to the match is the minimum of the two attached weights.

The counts attached to the fragment of six carbon atoms produced by long carbon chains tended to overwhelm more important parts of the molecules, causing unwanted matches. To circumvent this problem, the multiplicity weight was modified for compounds with long carbon chains. Thus, the weight attached to the six-carbon-chain fragment is the log to the base 2 of (the multiplicity plus 1) to the next whole



Fragment number	Multiplicity	Weight: Size	Frequency	Total
1	1	1	4	4
2	1	1	6	6
3	1	1	5	5
4	1	1	4	4
5	1	1	4	4
6	1	1	2	2
7	1	2	1	2
8	1	2	4	8
9	2	2	1	4
10	1	2	1	2

Figure 2. Typical compound from our file, its fragments, and their weights.

number. These weights are shown in Table II.

As a variation, all the weights from multiplicity times size were reduced the log to the base 2 as in Table II. Thus, each new copy usually receives less additional weight.

(2) Size Weights. An early idea was to weight the fragments according to the number of atoms, something like molecular weight. However, just the reverse is required. The reason is that the larger fragments contain several bonds, each the center of a new fragment. Thus, a difference in metal atoms would be outnumbered by partially redundant overlapping fragments. Even small functional groups would sometimes be subordinated to the larger structure. Therefore, the plain large fragments are unweighted (i.e., weight 1), the special medium-sized fragments receive weight 2, and the metal atoms receive weight 4. Thus, the size weights are generally comparable in magnitude to the multiplicity weights. As mentioned earlier, an alternative weighting scheme reduces the size weight times the multiplicity weight by the logarithm as in Table II.

(3) Frequency Weights. The fragments also vary greatly in incidence among the compounds. It is evident that the lower incidence, more distinctive, fragments should get higher weight. Therefore, a frequency count was taken for all the fragments. We used a frequency weight with the general formula 4 − log (I), where I is the incidence. Exhaustive generation implies that I is never 0. This weight varies from 1 to 4, the same range as the size weights, upon appropriate choice of the base of the logarithm. For the 4980 compounds the base 10 works well since the highest incidence for a fragment is just about 1000. For the larger set of compounds, the base must be increased.

An alternative frequency weight that gave more weight to the rare fragments seemed to work better (as described later). This weight ranges from 1 to 10 and decreases by 1 with each doubling of the incidence of the fragment. The formula is $\max[1,10 - \log_2 (I)]$.

(C) Similarity Measure. Thus, each fragment gets a weight incorporating its size, its frequency, and its multiplicity; the last is the only factor that can vary among compounds. These weights, assigned to each distinct fragment in a compound, are added together to constitute the total weight for the compound.

The similarity measure used for matching two compounds was computed by adding the weights of the fragments common to both compounds and dividing the sum by the weight of the larger compound. As was mentioned earlier, when matching fragments differ in weight, the minimum of the two weights yields the weight of the fragments in common. Let $w(i,j)$ represent the weight of the ith fragment in the jth compound. Suppose j has less total weight than k. The similarity measure is then

$$\text{sum } \min[w(i,j),w(i,k)]/\text{sum } w(i,k)$$

The measure in Willett most closely related to our similarity measure is the Canberra distance. Though there was a slight preference for the Tanimoto measure, Willett's analysis of six measures did not show any clear superiority for any measure. We can illustrate a problem with the Tanimoto and cosine measures, where the numerator is the sum of the products of the weights. The Tanimoto measure is

$$\text{sum } w(i,j)w(i,k)/\text{sum } w(i,j)^2 + w(i,k)^2 - w(i,j)w(i,k)$$

The cosine measure is

$$\text{sum } w(i,j)w(i,k)/[\text{sum } w(i,j)^2 \text{ sum } w(i,k)^2]^{1/2}$$

These measures give too much weight to already highly weighted features. For example, consider two compounds, j and k, each with six fragments, but only one fragment in common and the following weight vectors:

$$j = (10,1,1,1,1,1,0,0,0,0,0)$$

$$k = (10,0,0,0,0,0,1,1,1,1,1)$$

The Tanimoto measure is 100/110 on a scale of 0 to 1, and the cosine measure is 100/105. The angle between the two vectors is most strongly influenced by the larger components with less regard to the dimensionality. By our similarity measure the match is 10/15.

Thus, we see that if the high weight of a common feature is due to multiplicity of a fragment rather than a rareness factor, then we would not want the compounds to match so closely. This provides motivation for reducing the size–multiplicity weight by the logarithm and using a significant rareness factor. Empirical optimization of alternative weights will be discussed after the clustering algorithm is introduced.

Note that two different compounds can sometimes have identical sets of fragments and weights. This is obvious for compounds with long carbon chains mentioned earlier. For a more interesting example from the data see Figure 3. Such pairs of compounds cannot be separated under this system of

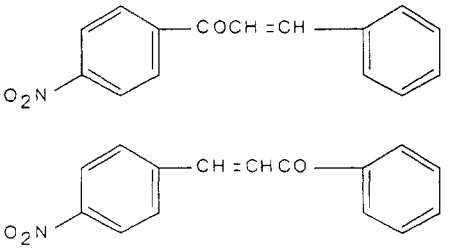


Figure 3. Two compounds from our sample that had identical sets of fragments.

fragments regardless of the clustering algorithm. For smaller fragments, e.g., those used in Willett's work, such conjunctions would be more extensive.

CLUSTERING ALGORITHM

A simple clustering algorithm called the leader algorithm[6] was used. This algorithm has several advantages for the chemical clustering application. It does not require simultaneous determination of a similarity measure for all pairs of compounds. This would be not too difficult for the 4980 compounds but practically impossible for the 232 000 compounds that are planned. It would also be prohibitive to find nearest-neighbor lists for all these compounds.

Compounds, sorted on increasing total weight, are entered one at a time. If a newly entered compound does not match any compounds on the current list of leaders, it is added to the list as a new leader. At the start the list of leaders is empty. If the entering compound matches any leader, it is added to that cluster and proscribed from becoming a leader. Thus, each leader determines a cluster of one or more compounds. See Figure 4 for a flowchart of the algorithm.

Ordering of compounds by increasing total weight arranges that the leader of each cluster will be a compound of lowest total weight in that cluster. This simplifies analysis of the results. Note that each compound will have a chance to match all leaders of lower weight. Moreover, since every compound in any cluster has successfully matched with the leader, the leader becomes the likely candidate to sample from each cluster.

One can see that a weakness of the leader algorithm is its dependence on the order of entering compounds. As we have just seen, this property can be used to advantage. On the negative side, two compounds that match very closely may not get into the same cluster if one of them is picked up by a leader that just misses matching the other. However, such a split can happen in other clustering methods, for example, the more sophisticated Jarvis–Patrick[7] method.

When the leader algorithm is optimized for speed, the entering compound is assigned to the earliest matching leader. A modification for accuracy assigns it to a closest matching leader.[8] Since we allow overlap, we assign it to all matching leaders. In all three cases there are the same number of leaders and therefore clusters.

Note that the list of leaders is formed ordered by compound total weight. We do not have to compare entering compounds with leaders that are too small to meet the matching criterion, even if all the leader's fragments are contained in the entering compound. As the clustering progresses, we keep a record of total weight vs initial points on the leader list. This enables us to save time by always comparing only with leaders in the appropriate top end of the list. In practice this allowed us to compare entering compounds with, on the average, about one-third of the leaders.

OPTIMIZATION

Optimizing the clustering was rather pragmatic. There were several parameters to set. The main parameter was the required fraction of fragments in common for two compounds to match. This was set at 0.65 of the larger molecule after it was noticed that at least 0.64 was required to separate single from double alcohols.

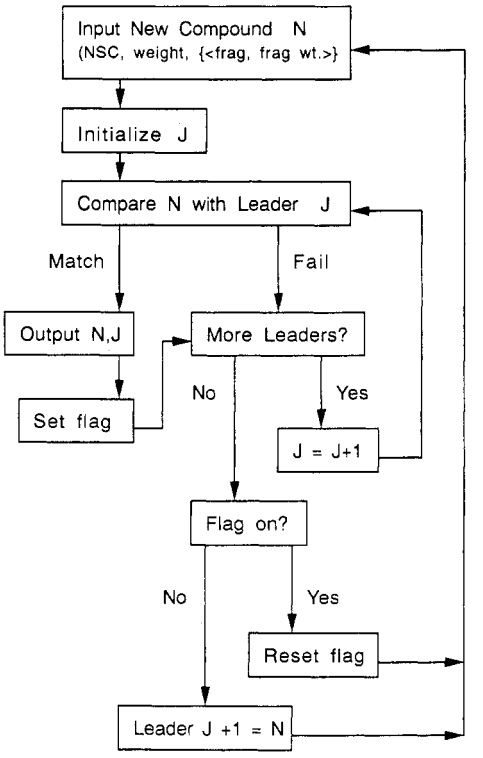


Figure 4. Flowchart for our version of the leader algorithm, allowing overlapping clusters.

Other parameters were set by comparing the overlapping clusters produced by the various alternatives. Overlapping clusters simultaneously show the extent and the limitations of matching, since the leaders of a pair of overlapping clusters are obviously related compounds that do not match.

There was also a general goal of decreasing the amount of overlap. This decrease signals a better classification of the compounds. For example, when the compounds were ordered by increasing age, i.e., a reverse chronological ordering, instead of by weight, there was an enormous increase in overlap. This can be explained as follows. Suppose the ordering by weight yields a cluster L, LA, LB, ... with leader L and LA does not match LB. Then if LA and LB occur before L in the age ordering, L will fall in both clusters. We think the ordering by weight yields a more natural, less arbitrary, classification.

There were two reasons for using the weight of the larger compound rather than the sum of the weights of both compounds, as the denominator of our similarity measure. The latter denominator would cause our measure to be effectively the same as the Canberra distance in Willett's book. The original reason for using only the larger compound was to facilitate matching when the (larger) candidate compound was closer in size to the leader. The second, confirming, reason resulted from a comparison run using 0.35 of the sum of the weights instead of 0.65 of the larger weight. The change produced more clusters and more overlap. Decreasing from 0.35 to equalize the number of clusters would further increase the overlap. This results in a poorer classification.

Optimizing weightings was a bit more complicated. A systematic comparison was undertaken of two alternatives in size–multiplicity weighting and two alternatives in frequency weighting. The results are shown in Table III. Overlap is measured by the number of occasions compounds match more than one leader.

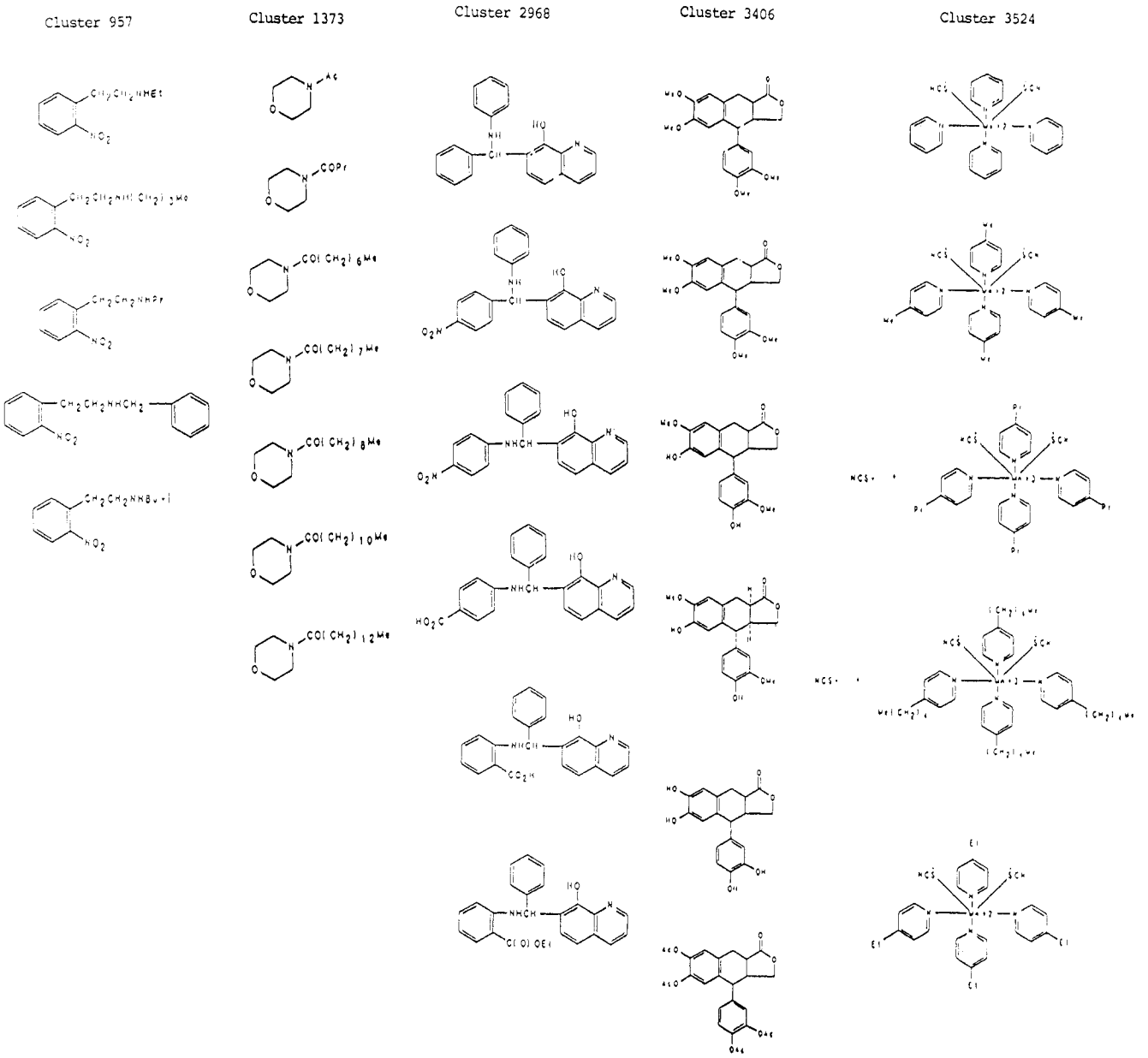


Figure 5. Some of the larger, more interesting, clusters.

We see from Table III that when the frequency weight ranges from 1 to 4, the size–multiplicity weight that applies the logarithm to carbon chains gives better results than uniform application of the logarithm. This may be due to compensation of high multiplicities forcing distinctions that are lacking from purely frequency, i.e., rareness considerations. The results are mixed when the frequency weights range from 1 to 10. Here, there are six more clusters under uniform application of the logarithm but nine less occurrences of overlap.

This last alternative was chosen as the more naturally appealing situation, each new copy of a fragment usually having less weight as they accumulate. Also, the extremely low overlap shows good separation of classes.

Moreover, a run of a 50 000-compound sample under the former conditions showed a large increase in overlap, roughly 2000 extra matches. Therefore, the latter optimum was chosen as a means of keeping overlap down in the larger run. Results on the larger runs will be reported in the next paper.

RESULTS AND CONCLUSIONS

Aside from the large size of the data, there are two striking aspects of this work. The first is the sparseness of the clustering. Of the 4980 compounds, 3053 failed to match with any others, and the remaining 1927 compounds formed 732 clusters. This shows that classification of chemical structures is not always a practical thing to do. The compounds in our file were especially selected for diversity of random screening. In our work these odd structures are important, and we do not force them to cluster. See Figure 5 for some of the more interesting larger clusters.

The other aspect runs in the opposite direction. The techniques of clustering arise from biological and other sciences where objects fall into disjoint classes. However, chemical classes based on molecular structure are not always so well-defined. In fact, one can almost arbitrarily synthesize compounds to force overlap of classes, e.g., a nucleic acid mustard. We allow overlap of clusters when the matching criteria are met, so that similar compounds are easily detected.

To be precise we should call this work "clumping" or overlapping sets rather than "clustering". However, overlap is not extensive under our strong requirement for matching. There is also the large number of compounds that do not form clusters, let alone clumps.

Of the 4980 compounds, 21 landed in more than one cluster, affecting 36 clusters. We show an example of a cluster that

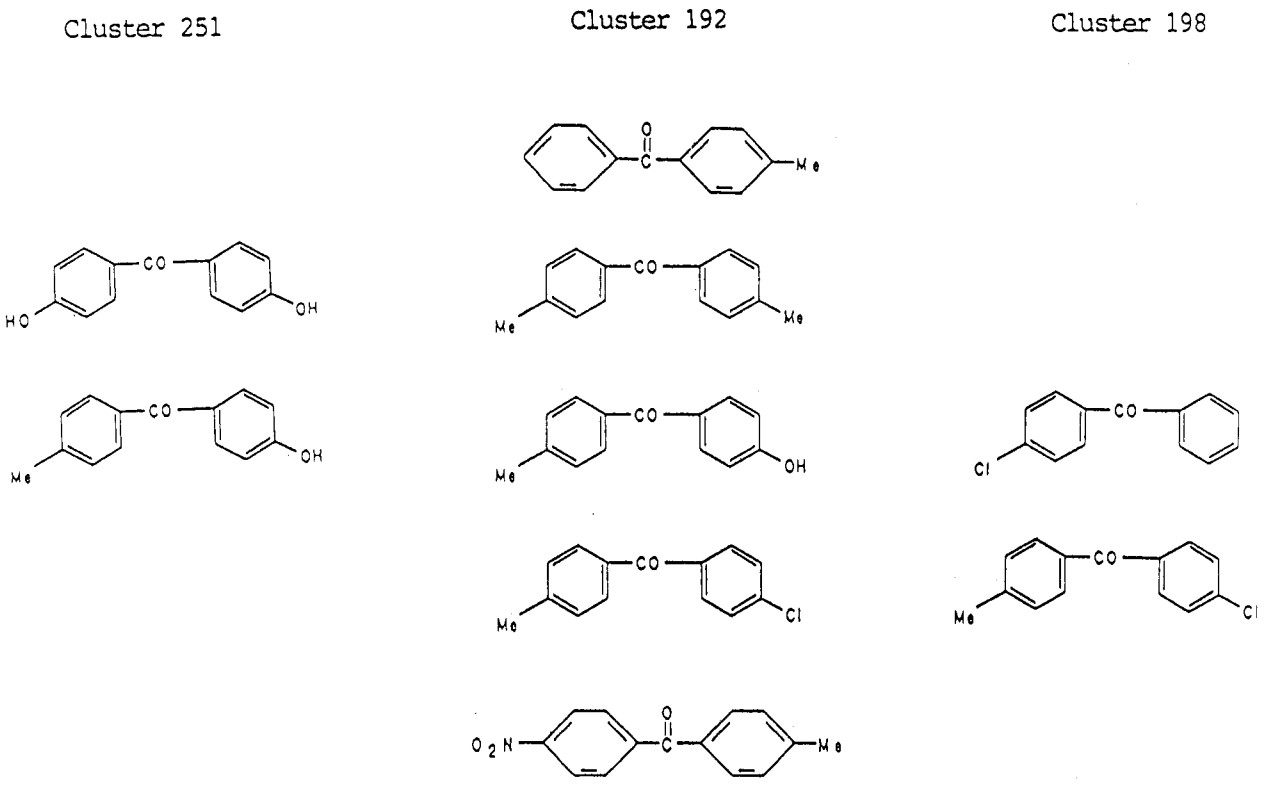


Figure 6. Three typical clusters, where the center cluster has a compound in common with both side clusters.

Table III. Comparison of Alternative Fragment Weightings

	frequency wt			
	range 1–10		range 1–4	
size–multiplicity	clusters	overlap	clusters	overlap
log only C-chain	3779	30	3683	73
log on all fragments	3785	21	3705	80

overlapped with two other clusters in Figure 6.

Considering this work as a method of obtaining a representative sample, say, simply the leaders, we have reduced the data from 4980 to 3785 clusters, about a 24% reduction. The 50 000-compound run yielded 26 000 clusters, including 16 500 singletons. With the full 232 000 compounds, the percentage of singletons will be even lower, and the reduction of total clusters is expected to be well over 50%.

When screening 10 000–30 000 compounds per year, one can appreciate the differences between dealing with 232 000 or 75 000–100 000 compounds. The work reported here is new because of the integrated approach to this specific problem. For example, previous methods, as in Willett,[2] diminished singleton clusters by the adaptive scaling of the Jarvis–Patrick[7] method and/or assigning singletons to some nearest clusters. We welcome singletons as diverse compounds and expect them to represent about half the clusters in the final run.

Unfortunately, processing time for clustering can increase with the square of the number of compounds. The 4980 compounds took about 110 s of CPU time on the IBM 3090. Inverting the files as in Willett[9] produced runs of about 70 s. A version of the algorithm that continuously checked for nonmatches reduced the time to about 56 s, but it is not compatible with the earlier reduction. Thus, the 232 000 compounds can take as much as 34 h of CPU time. This seems quite prohibitive.

The time for running the leader algorithm is more closely related to the number of compounds times the number of leaders. A likely estimate of the number of leaders in the 232 000 compounds is 100 000. This would still require 20 h of CPU time. The next paper will show how this difficulty was resolved by the use of a massively parallel processor.

REFERENCES AND NOTES

(1) Boyd, M. R. National Cancer Institute Drug Discovery and Development. In Frei, E. J., Freireich, E. J., Eds. *Accomplishments in Oncology*; Lippincott: Philadelphia, 1986; Vol. 1, No. 1, p 8.
(2) Willett, P. *Similarity and Clustering in Chemical Information Systems*; Research Studies Press: New York, 1987.
(3) Willett, P.; Winterman, V.; Bawden, D. Implementation of Nonhierarchic Cluster Analysis Methods in Chemical Information Systems: Selection of Compounds for Biological Testing and Clustering of Substructure Search Output. *J. Chem. Inf. Comput. Sci.* **1986**, *26*, 109.
(4) Attias, R. DARC Substructure Search System: A New Approach to Chemical Information. *J. Chem. Inf. Comput. Sci.* **1983**, *23*, 102.
(5) Hodes, L. Selection of Molecular Fragment Features for Structure–Activity Studies in Antitumor Screening. *J. Chem. Inf. Comput. Sci.* **1981**, *21*, 132.
(6) Hartigan, J. A. *Clustering Algorithms*; Wiley: New York, 1975.
(7) Jarvis, R. A.; Patrick, E. A. Clustering Using a Similarity Measure Based on Shared Nearest Neighbors. *IEEE Trans. Comput.* **1973**, *C-22*, 1025.
(8) This version is similar to the single-pass clustering method described in ref 3, p 110. I think using the centroids of the clusters as Willett does would unduly complicate analysis and leave us without a natural representative in many cases.
(9) Ref 2, p 208.