LITERATURE CITED

(1) (a) This work was presented in part at the 7th Middle Atlantic Regional Meeting of the American Chemical Society, Philadelphia, Feb 14–17, 1972. For an earlier paper and leading references see W. T. Wipke, S. R. Heller, R. J. Feldmann, and E. Hyde, "Computer Representation and Manipulation of Chemical Information," Wiley, New York, N.Y., 1974. (b) Merck Career Development Award. Current address: Department of Chemistry, University of California, Santa Cruz, Calif. 95060. (c) NSF Trainee, 1969–71.

(2) Corey, E. J., and Wipke, W. T., *Science*, **166,** 178 (1969).

(3) Welch, J. T., Jr., *Assoc. Computing Mach.*, **13,** 205 (1966).

(4) Gotlieb, C. C., and Corneil, D. G., *Comm. Assoc. Computing Mach.*, **10,** 780 (1967).

(5) Paton, K., *Comm. Assoc. Computing Mach.*, **12,** 514 (1969).

(6) Tiernan, J. C., *Comm. Assoc. Computing Mach.*, **13,** 722 (1970).

(7) Fugmann, R., Dolling, U., and Nickelson, H., *Angew. Chem., Int. Ed. Engl.*, **6,** 723 (1967).

(8) (a) Long, P. L., Masters Thesis, Ohio State University, 1970; (b) Phares, R. F., and White, L. J., Ohio State University Computer Information Science Research Technical Report, OSU-CISRC-TR-70-7; (c) Long, P. L., Phares, R. F., Rush, J. E., and White, L. J., Abstract CHLT-15, 160th National Meeting of the American Chemical Society, Chicago, Ill., Sept 1970.

(9) Gibbs, N. E., *J. Assoc. Computing Mach.*, **16,** 564 (1969).

(10) Corey, E. J., and Peterson, G. A., *J. Am. Chem. Soc.*, **94,** 460 (1972); Plotkin, M., *J. Chem. Doc.*, **11,** 60 (1971); Bersohn, M., *J. Chem. Soc., Perkin Trans. 1,* 1239 (1973).

(11) Ring *i* is a member of the same assembly *k* as ring *j* if ring *i* shares one or more *edges* with ring *j*, or if ring *i* shares at least one edge with another ring previously found to be a member of assembly *k*. Note that sharing only a *node* (spiro linkage) between ring *i* and a member ring of assembly *k* is insufficient for inclusion of ring *i* in assembly *k*.

(12) An adjacency matrix is a square matrix where a 1 in the *i,j* position indicates atoms *i* and *j* are bonded (adjacent), while a 0 in the *k,l* position indicates atoms *k* and *l* are not directly bonded (not adjacent). In an incidence matrix, the column index refers to the bond while the row index refers to the atom, so that if the *i,j* position is a 1, atom *i* is on bond *j* and if the *k,l* position is a 0, atom *k* is not on bond *l*. See also M. F. Lynch, J. M. Harrison, W. G. Town, and J. E. Ash, "Computer Handling of Chemical Structure Information," American Elsevier, New York, N.Y., 1971.

(13) Morgan, H. L., *J. Chem. Doc.*, **5,** 107 (1965). The Morgan name is the canonical name used by the Chemical Abstracts Registry system.

(14) For a stereochemically unique naming algorithm, see W. T. Wipke and T. M. Dyott, *J. Am. Chem. Soc.*, **96,** 4834 (1974).

(15) Logical exclusive or gives all of the elements in either set, but not in both. The importance of set representations was first pointed out by C. H. Sussenguth, *J. Chem. Doc.*, **5,** 36 (1965).

(16) Without the use of sets of the "exclusive or" sets, the storage required is essentially twice as great.[7b] The use of the *sets of sets* also speeds up the algorithm considerably, since it replaces the physical transfer of the "exclusive or" sets with a fast operation on the set of the "exclusive or" sets.

(17) The structures given in the "Dictionary of Organic Compounds" and the 18th edition of the "Merck Index" are used as the justification for this statement.

An Efficient Design for Chemical Structure Searching. I. The Screens†

ALFRED FELDMAN*

Walter Reed Army Institute of Research, Washington, D.C. 20012

LOUIS HODES

National Cancer Institute, National Institutes of Health, Bethesda, Maryland 20014

Received January 21, 1975

A method has been developed for generating efficient screens for chemical structures. Fragments are generated by an algorithm under control of file statistics. The fragments obtained are normalized by weighting their code patterns. Superimposition of these codes yields the screen codes for the structures.

I. INTRODUCTION

The chemical structure search system at the Walter Reed Army Institute of Research (WRAIR) has been in operation since 1962.[1] In this period, its files have grown to about a quarter million compounds. Currently, the system is being converted, from a sequential tape file operation, to a direct access file based design. As part of this effort, an improved system of screens has been developed.

Screens are used in all files with substructure search capability. Their use makes it possible to avoid much laborious atom-by-atom matching: the more efficient the screens, the fewer the compounds that must be eliminated on the basis of atom-by-atom matching.

† Paper presented at the 167th and 168th National Meetings of the American Chemical Society, Atlantic City, N.J., Sept 1974, and Philadelphia, Pa., April 1975.

Screening can be thought of as a conventional search, in which structural fragments are used as descriptors for compounds and queries. Where these descriptors cannot be matched, further searching is unnecessary. Screening is thus related to the classification problem, where it is asked what are the best attributes for classification, and how many should there be. Compared with nonchemical data files, collections of chemical structures are remarkable in that their attributes—generally their fragments—are precise and abundant. This abundance, evident from the example shown in Figure 1, forces a choice among fragments and introduces the problem of optimizing screens for searching.

As yet, no system has advanced a set of optimum screens. Chemical files, consequently, are partly underscreened and partly overscreened. They cannot be efficiently searched nor, for that matter, efficiently stored. There is not even agreement as to the approach to be taken toward screen

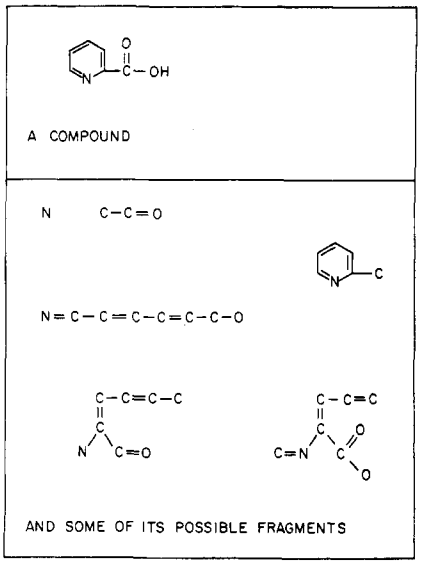


Figure 1. From this compound, a hundred fragments can be obtained. Since single as well as multiple occurring fragments can be used as screens, the number of possible screens is still higher.

generation, the "intuitive" methods holding their own against the "algorithmic" ones. Because of the extensive processing generally required by chemical systems, this lack of efficiency can easily lead to an abuse of resources.

For the formal evaluation of a set of screens, all interactions among the screens would have to be considered—a very tedious process. Yet the problem of generating optimum screens goes beyond the selection of the best from a given set; it requires the selection of the best from among all possible screens. Given that a simple compound, such as the one shown in Figure 1, already can yield over 100 fragment screens, the number of possible screens must truly be astronomical.

II. THE GENERATION OF SCREEN FRAGMENTS

1. The Strategy. In attempting to develop improved screens for the new WRAIR system, the approach developed was based on a strategy used in artificial intelligence and game theory. A problem posed there is the determination of the outcome of a game. In any but the simplest games, the number of choices involved in the sum of moves and countermoves is astronomical. Representing the moves of a game as the branches of a genealogic tree, the strategy aims at the early removal of as many branches as possible. For each branch grown in a generation, the prospects of winning are evaluated. If deemed unsatisfactory, the branch is pruned. Pruned branches can grow no further, and this eliminates from consideration a great many future moves. If this "forward pruning" keeps pace with the growing, the problem becomes tractable. Though this strategy may not be optimum, it may lead to satisfactory results.[2]

In our adaptation of this strategy, structural fragments were grown in the manner of the above trees. The process begins with single atoms, then becomes iterative. In each iteration, a generation rule is used to add single atoms to fragments passed on from the last iteration. In this manner, evidently, all possible fragments can be generated (Figure 2). In each iteration, however, a selection also takes place. An elimination rule determines which fragments are to be pruned, and which fragments are to be passed on to the next iteration, where they will grow further.

In order to keep within reasonable bounds the number of fragments to be processed by these two rules, a heuristic was used additionally.

2. Growing and Pruning the Trees. There being little

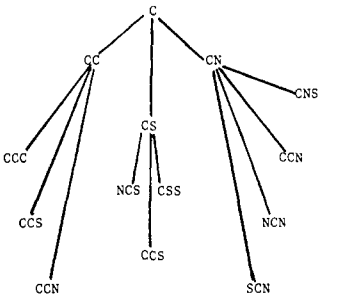


Figure 2. Generation of fragments by adding one atom to a predecessor fragment. Note that the fragment NCS obtained in the second lineage is identical with the fragment SCN obtained in the third. A fragment thus may have more than one parent.

hope of obtaining an elimination rule taking into account mutual interactions among fragment screens, the growing and pruning of fragments was done on the basis of their file incidence.

One can conceive of a file as being made up of overlapping bodies of compounds, each body corresponding to a fragment's incidence set. When growing a fragment, several successors are normally obtained, of which each normally occurs with an incidence lower than that of its parent. The objective being to break up the large compound bodies, we used the high incidence fragments, selectively, as a source for growing new fragments. This procedure was repeated until enough discrimination was obtained.

Each iteration of this process produces also a large number of low incidence fragments. Were these allowed to grow further, file bodies would be broken up that are already very small. Consequently, the low incidence fragments were pruned. This precludes the formation of a large number of redundant screens.

The fragments generated were obtained from a file containing initially a 10% sample, or about 26,000 compounds, of the entire WRAIR file. At each iteration, incidence counts were taken of all successor fragments obtained. Fragments with an incidence above a cutoff point, for which a value of 1% was chosen, were destined for additional growth. Fragments occurring with low incidence, i.e., below another cutoff point, for which a value of 0.1% was chosen, were pruned. This was accomplished by removing, from the sample file, the compounds containing them. Additionally, compounds not containing high-incidence fragments were also removed. The compounds remaining in the file afterward had their fragments grown in the next iteration.

A complete iteration is diagrammed in Figure 3. The reduction of the sample file size, as a consequence of pruning, is shown in Figure 4.

3. The Heuristic. As mentioned, a heuristic was used in addition to pruning. Its purpose was to exclude, a priori, the large number of fragments whose structures are either chemically unlikely or too complex. The latter were avoided for the pragmatic reasons of simplicity of programming and speed of processing.

The heuristic was based on the experience obtained with the screens of the early WRAIR system[1] and on the work published by the Sheffield group.[3] Broadly speaking, the eliminations became more severe with increasing fragment size, an exception being made for benzene rings and their substituents. Thus, in generation 1, multiple occurrences of atoms and, in generations 6 and 7, multiple occurrences of substituted benzene rings were allowed. In generations 1 and 2, the more common atoms (C, N, O, S) were differentiated according to bonds attached and valence exhibited. Thereafter, that distinction was no longer made. Generation 3 allowed all "augmented" atoms, i.e., groups consisting of central atoms and their attachments. Thereafter,

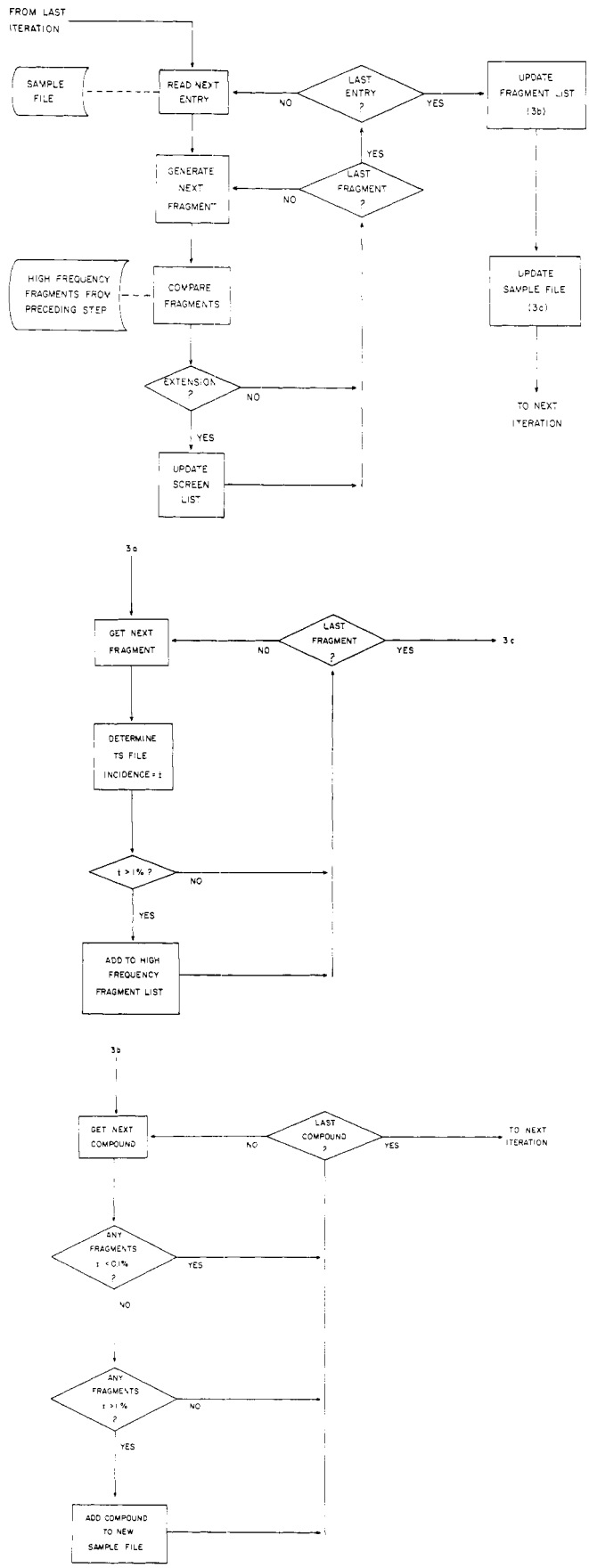


Figure 3. (a) One iterative step of the screen generation; (b) update of fragment list; (c) update of sample file.

fragments were limited to nonbranching chains or rings, with rings, furthermore, allowed only a single chain attachment. Following generation 8, only fragments consisting of single or double rings, with or without single chains attached, were allowed, but no chains without rings.

Generation	File Size
1	26,470
2	26,183
3	25,179
4	24,389
5	23,730
6	22,889
7	21,543
8	19,590

Figure 4. Reduction of file size as a result of pruning. Following the 8th iteration, the file was no further reduced.

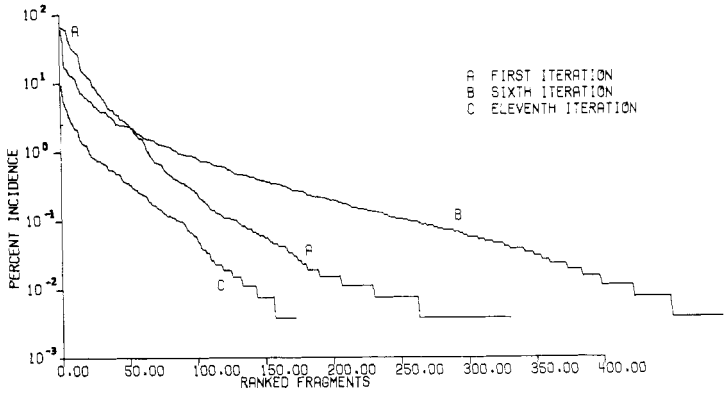


Figure 5. Number of fragments generated in the first, middle, and last iterations. The scales used here emphasize differences in the tails of the graphs.

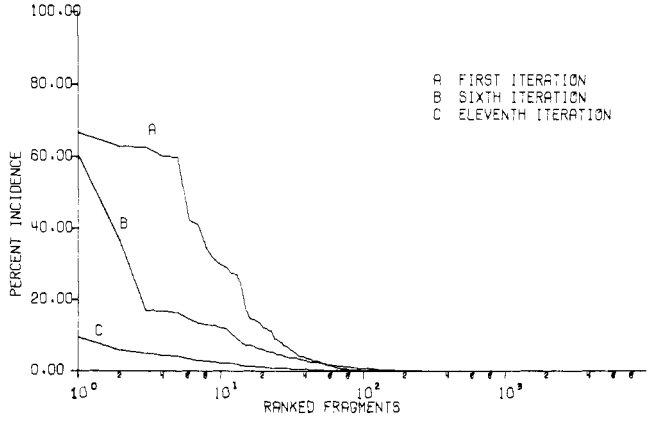


Figure 6. Number of fragments generated in the first, middle, and last iterations. The data here are the same as in Figure 5. The different scales emphasize differences in the peaks of the graphs.

4. The Screens Obtained. Figures 5 and 6 show the distribution of the fragments following the first, middle, and last iterations. The graphs were obtained by plotting, in the manner of the Sheffield group,[3] the incidence of the fragments obtained vs. the fragments ranked by incidence. The tail of the graph thus represents the low incidence fragments, and the peak those with high incidence.

Figure 5, which uses a linear scale for the x axis, illustrates the fate of the low incidence fragments. The tail of the graph, moderate following the first iteration, grows considerably following the sixth. Evidently, the successors to the fragments formerly in the peak have migrated to the

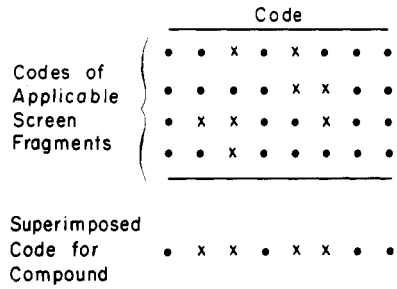


Figure 7. Assembly of a superimposed code using, for the sake of illustration, an 8-bit dedicated space.

Fragment	Freq.	First-order Discrimination	Second-order Discrimination	Bit code	Assigned bits
Zn	68	3	3	oolollooo	oolollooo
S-Zn	33	4	1	olooooooo	ollollooo
C-S-Zn	27	4	0		ollollooo
N-C-S-Zn	21	4	1	ooolooooo	olllllooo
S-C-S-Zn	19	4	1	ooooooool	ollollool
C-C-S-Zn	7	6	2	oooloolooo	olllllloo
O-C-S-Zn	2	7	4	looloooll	lllllloll

Figure 8. Determination of the number of bits assigned to the fragment screens. The screens shown were obtained with the frequencies indicated. The number of bits computed has been reduced to simplify the illustration

tail. Following the eleventh iteration, the tail has almost disappeared, a result, obviously, of steady pruning.

Figure 6, which shows the same data using a linear scale for the y axis, illustrates the fate of the high incidence fragments. The peak of the graph, following the first iteration, is tall, denoting the high incidence fragments present. Following the sixth iteration, the peak is still tall, but somewhat thinner. Following the eleventh iteration, it has descended to about 10%.

Altogether, the iterative procedure was repeated 11 times and provided 3758 fragment screens, ranging in size from one to eleven atoms. The iterative procedure might have been repeated an additional number of times, eliminating the distribution curve altogether, but this was not deemed worthwhile.

III. CODING THE FRAGMENT SCREENS

1. The Method. The fragments obtained above do not yet constitute very good screens. This is due, in part, to the variation in their incidence, which ranges from about 70% to less than 0.1%, and, in part, to their remaining redundancy. Although many redundant screens were avoided as a result of pruning, considerable redundancy remains among the fragments obtained. This is because most of them are separated from a parent by a distance of only one atom. Thus, whenever a screen is applicable to a particular compound, all its predecessors apply to that compound as well. This has led us to develop a mechanism to compensate for both this redundancy and the variation in incidence.

The compensation is implemented in the process of assigning codes to the fragment screens. Other systems code their screens on a one-to-one basis. That is, each screen is given the same importance, regardless of its power to discriminate. Our compensating mechanism operates by weighting the fragment screen codes.

As a coding method that would lend itself to the desired weighting, we selected the so-called "superimposed code" which Mooers[4] developed a number of years ago for punched cards (Figure 7).

In this method, each of a set of attributes is identified by a unique bit pattern. The code of a particular entity is obtained by superimposing the codes of the attributes possessed by that entity. In our application, the fragment screens obtainable from a chemical compound are considered its attributes.

The method of superimposed coding can be said to have three parameters. One is the layout of the bits (i.e., the distinctive pattern assigned to each attribute), the second is the number of one-bits constituting that pattern, and the third is the amount of "dedicated space" (equal for all attributes) into which the patterns are placed.

2. Determination of the Number of Bits Assigned to a Fragment Screen. We began by determining the number of one-bits to be assigned to each fragment. Our objective was to use the freedom to vary this number for compensating for the difference in incidence of the screen fragments. For example, if the required number of bits is determined by formula

$$B = \log_2 1/p \tag{1}$$

where B = number of bits to be assigned to screen and p = incidence of screen (where $0 \leq p \leq 1$), this number will be proportional to a screen's first-order information theoretic importance, in the sense given to that term by Shannon.[5] The weighting thus could exactly balance differences in discrimination.

That weighting, however, would be correct only under statistical independence of the screens. To correct for the redundancy among screens, the number of bits assigned to each screen fragment was computed, by formula 2, on the basis of the difference between a screen's incidence and that of its nearest parent.

$$\text{No. of one-bits/screen} = \log_2 \frac{\text{incidence of parent}}{\text{incidence of screen}} \tag{2}$$

This correction, which we call the second-order discrimination,[6] does not consider associations other than the nearest predecessor in file incidence (cf. Figure 2) and, consequently, some redundancy remains. The major amount of redundancy, however, has been removed.

Figure 8 illustrates the bit number calculation. Seven fragments are shown. The first three (column 1) are predecessors of each other; the last four are successors of the third fragment (C-S-Zn). Column 2 shows their respective incidences, column 3 the number of bits calculated for them by eq 1, and column 4 the number of bits calculated for them by eq 2. A code, with the correct number of bits for each fragment, is shown in column 5. The cumulative bit codes assigned to a compound are shown in column 6. A closer look at these codes will show that, for any fragment assigned to a compound, the predecessor screens of that fragment are also assigned. It will be seen further that these codes correspond, in number, to the bits required by eq 1 (column 3). The small discrepancies are the result of roundoff. The codes obtained in column 5 are thus weighed to compensate both for the variable incidence of their fragments and for their redundancy.

As the result of zero-bit assignments, 563 screens were eliminated, which reduced their total to 3195.

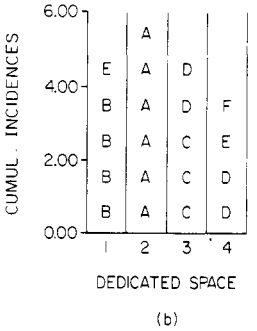
3. Determination of the Dedicated Space. Next, the required amount of dedicated space was determined. Our objective was to arrive at an average bit density of 50% which, according to a principle stated by Mooers,[4] represents the maximum efficiency attainable for superimposed codes. Having computed, by the above procedure, the number of one-bits to be assigned to each of the 3195 fragments obtained, the amount of space required to produce a 50% density could be calculated from the equation

$$D = \sum_{i=1}^{n} (s_i f_i)/Mc \tag{3}$$

Here, the number of bits (s_i) assigned to each screen is multiplied by the incidence (f_i) of the screen; the products are added and divided by the number of compounds (c) in

SCREEN	#BITS	INCIDENCE	PATTERN
A	1	6	0100
B	1	4	1000
C	1	3	0010
D	2	2	0011
E	2	1	1001
F	1	1	0001

(a)



(b)

Figure 9. Determination of the bit pattern for the fragment screens. The screens in column 1 of the table (9a) are assumed not to be related. The number of bits to be assigned to each is shown in column 2. The incidence of each screen is shown in column 3. A dedicated space of 4 bits is assumed for the sake of illustration. The height of the boxes in the diagram (9b) is proportional to the screen's incidence. The height cannot exceed the Mooers level (see text). The corresponding pattern is shown in column 4.

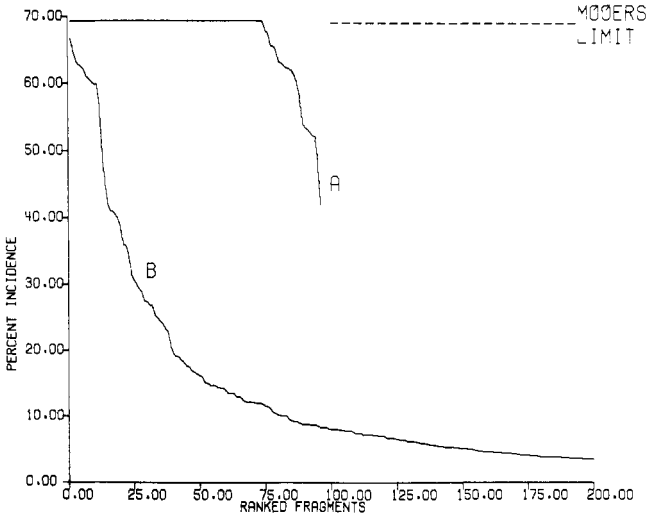


Figure 10. Comparison of the distribution curve for fragment screens (B) and bit screens (A). To allow use of the same scale, the curve for the fragment screens has been truncated. It extends actually beyond 3,000.

the sample file. The resulting average number of one-bits is divided by Mooers' compression factor M, which represents the loss of one-bits incurred from superimposition, and whose value is 0.69, given a 50% one-bit density.[4]

The amount (D) of dedicated space so computed was 93 bits. This is very close to the convenient number 96, which represents 4 words on our machine (CDC 3500) and 3 words on a 32-bit/word machine. The dedicated space allocated to the new WRAIR screens was consequently set to 96 bits. (Using the dedicated space of 96 bits, the compression factor was recomputed to 0.67; thus the average one-bit density of the slightly expanded dedicated space will be 0.49.)

4. Determination of the Distinctive Patterns. Finally, the third parameter, the distinctive patterns that characterize each of the fragment screens, was obtained. The one-bits allocated to each screen fragment were assigned individually, and at random, over the dedicated space. Fragments were taken in order of decreasing incidence. In the assignment, identical locations were avoided for bits originating from the same fragment or from the nearest parent. At each location, cumulative totals of incidences were kept, and the Mooers "limit," i.e., a total incidence of 0.69, was not exceeded (Figure 9). In the case of conflict, the next generated random location was assigned.

The distribution curve of the bits so assigned is shown in Figure 10A. It is seen that, for the major part of its length, this curve espouses the Mooers limit, which is the bit distribution that will result in a superimposed code of 50% density. For comparison, a portion of the distribution curve obtained for the 3195 fragment screens is also shown (Figure 10B).

IV. EVALUATION

One consideration in evaluating screens is their compactness. Having placed all screen codes within a dedicated space of 96 bits, the WRAIR system uses, effectively, only 96 functional screens for file discrimination. The 3195 fragment screens, obtained above, merely represent an intermediary stage.

96 bits is a considerable saving over the number of screens reported by other systems. A recently published system, which obtains its screens algorithmically, and whose holdings are of about the same size as those of the WRAIR system, carries over 10,000 screens.[7] The old WRAIR system used 500 "intuitive" screens.

Because of their compactness, the new screens will be economical of index storage. They will, also, not overburden a direct access system, so that it is not necessary to organize the screens into hierarchies,[8] which then impose a bias on searches.

The efficiency of the new screens further has facilitated the design of a file organization suitable for both identity and substructure searches.[9] In at least one current system,[10] separate methods are used for these purposes; this engenders a concomitant increase in the complexity of programming, housekeeping, and searching.

A second consideration in the evaluation of screens is the selectivity obtained. Our work was undertaken primarily because of the shortcomings of earlier screening methods. These, as mentioned, were of two types: the "intuitive" and the "algorithmic". In the intuitive systems, each screen is selected on its own merit—according to a chemist's best judgment—and the presence of each screen, in an input compound, must be ascertained more or less independently. These methods thus are slow. Our experience further had shown that some queries would grossly underscreen the file of the old WRAIR system, whose screens were intuitive. There was no indication that other intuitive screens would perform better.

The algorithmic systems also had poor selectivity. Being exhaustive, they produced good as well as ineffective screens, with the latter, unfortunately, greatly outnumbering the former. The reduction in computation time achieved by this approach was thus at the expense of the overall quality of the screens.

To compensate for the lower quality of the algorithmic screens, their number had to be increased. For example, the number of TSS screens,[11] which admit any acyclic subchain, is enormous. Clearly, this number cannot be allowed to increase unchecked. Most of the Sheffield screens[12] were consequently limited in size to just pairs with attached bonds, while most of the CAS screens[13] were limited to augmented atoms. Such simplistic limitations affect good and poor screens alike. Milne[11] pointed out, for instance, that the N–C–C–O grouping, which is important in α-amino acids, is not effectively screened when fragments are restricted to pairs of atoms, or to augmented atoms. Such deficiencies are not easily remedied ad hoc. In one system, separate algorithms were devised for different classes of screens, namely acyclic, single ring, and ring system nucleus, but, even then gaps remained.

It is, therefore, not surprising that underscreening and overscreening remained a problem with algorithmic screen systems and that their performance has been worse, at times, than that of screen sets based on intuition.

Our approach avoids the shortcomings of the earlier methods by evaluating the screens obtained. Having eliminated many redundant screens, it can accommodate both a larger proportion of good screens, and larger size fragments. As it is based on exhaustive file statistics, our approach may be expected to assign a generous complement of screens to virtually any query. Of course, this cannot be verified until after the system has undergone extensive utilization.

On two counts, our approach has been criticized. One is that, having stopped the iterative process after 11 generations, no fragments having more than 11 atoms were produced. There are, consequently, no screens with three or more fused rings. This is due to our fear that screens, obtainable beyond the 11th iteration, might be more troublesome than useful and that, when in doubt, initial underscreening was preferable to overscreening. But we shall monitor the performance of the system and, if need be, run a few additional iterations of the fragment generating algorithms and extend the heuristic to exclude, for example, the envelopes of multi-ring systems. Since incidences are low beyond the 11th generation, we do not expect having to expand the 96-bit dedicated space of the superimposed code.

The other criticism is that, by attuning our screens to file statistics, the efficiency of the screens will deteriorate as the composition of the file changes with time. True enough. But the holdings of a large file represent a considerable amount of inertia, and it is not anticipated that the screens would have to be regenerated more often than once every several years.

As our approach makes use of superimposed codes, another consideration is pertinent. With superimposed codes, false drops are inevitable. That is, sets of screens, different from that of a query, can accidentally yield a superimposed code pattern that includes the query superimposed code. However, as Mooers has pointed out,[4] the number of falsely retrieved entries can be estimated. Given the random assignment of one-bits, and the 50% density, each one-bit in the query superimposed code cuts the number of false drops by a factor of 2. This means, for example, that a fairly general query, with a superimposed code containing 18 one-bits (and thus 78 zeroes), will result in only one false drop in 256K, which is negligible. False drops, of course, are eliminated in the subsequent atom-by-atom search.

A final consideration in the evaluation of a screen system is the time required to obtain a complement of screens. With "intuitive" systems, the time, as stated, was long. With the algorithmic approach, which uses relatively few and simple rules, the time is much reduced. Although the rules used to generate the new WRAIR screens are more complex than those of earlier algorithmic systems, preliminary findings indicate a range of search time comparable to that of other algorithmic systems.

V. CONCLUSION

In conclusion, we have demonstrated that efficient screens can be generated automatically, that these can be obtained in a tractable number, even for on-line access, and that such screens are compact enough not to require different systems for identity and substructure searches.

Having implemented our programs on a CDC 3500, we have also demonstrated that they are usable on medium-sized machines.

We have further shown that by substituting a bit screen for the usual chemical fragment screens, considerably greater compactness and ease of use can be achieved while retaining substantially the discrimination of the original fragments.

As reported elsewhere,[9] a new file arrangement was designed for the WRAIR system. We believe, however, that the screens described here may be used with advantage for any filing system in current use, either one with inverted lists or one in which screens are sequentially scanned.

ACKNOWLEDGMENT

Programming support by A. A. Taoras and W. M. Waring is gratefully acknowledged.

LITERATURE CITED

(1) Jacobus, D. P., Davidson, D. E., Feldman, A. P., and Schafer, J. A., "Experience with the Mechanized Chemical and Biological Information Retrieval System," *J. Chem. Doc.*, **10**, 135–140 (1970).

(2) Simon, H. A., "The Sciences of the Artificial," MIT Press, Cambridge, Mass., 1970, pp 64 ff.

(3) Adamson, G. W., Lynch, M. F., and Town, W. G., "Analysis of Structural Characteristics of Chemical Compounds in a Large Computer-based File II. Atom-centered Fragments," *J. Chem. Soc., C*, 3702–6 (1971).

(4) Mooers, C. N., "Zatocoding Applied to Mechanical Organization of Knowledge," *Am. Doc.*, **2**, 20–32 (1951).

(5) Shannon, C. E., "A Mathematical Theory of Communication," *Bell Syst. Tech. J.*, **28**, 59 (1948).

(6) Hodes, L., "Selection of Descriptors According to Discrimination and Redundancy," in preparation.

(7) Lefkovitz, D., "The Large Data Base File Structure Dilemma," *J. Chem. Inf. Comput. Sci.*, **15**, 14 (1975).

(8) Feldmann, R. J., "Interactive Graphic Chemical Structure Searching," in "Computer Representation and Manipulation of Chemical Information," W. T. Wipke, et al., Ed., Wiley, New York, N.Y., 1974.

(9) Hodes, L., and Feldman, A., "An Efficient Design for Chemical Structure Searching. II. A Solution to the Large Data Base File Structure Dilemma," in preparation.

(10) Lefkovitz, D., Hill, H., and Hirschfeld, L., "National Cancer Institute's Drug Research and Development Chemical Information System: System Design," paper presented at the 169th National Meeting of the American Chemical Society, Philadelphia, Pa., April 1975.

(11) Milne, M., Lefkovitz, D., Hill, H., and Powers, R., "Search of CA Registry (1.25 million compounds) with the Topological Screens System," *J. Chem. Doc.*, **12**, 183–189 (1972).

(12) Adamson, G. W., Cowell, J., Lynch, M. F., McLure, A. H. W., Town, W. G., and Yapp, A. M., "Strategic Considerations in the Design of a Screening System for Substructure Searches on Chemical Structure Files," *J. Chem. Doc.*, **13**, 153–157 (1973).

(13) Wigington, R. L., "Machine Methods for Accessing Chemical Abstracts Service Information," Proceedings of the IBM Scientific Computing Symposium on Computers in Chemistry, 1969.