of searching quality on different substructure searching systems, though with the proviso that care must be taken to ensure that differences in graphical, or other, query input language do not invalidate the comparisons.

CONCLUSIONS

The effectiveness measure described above is a novel approach to the quantitative assessment of screen-set performance. It overcomes some shortcomings of the screen-out measure, particularly in the context of an operational system; the two measures may be used together for the most informative summary of screening performance. The measure has proved valuable in screen-set design for a connection table based graphical searching system and is suitable for long-term monitoring of system performance. It may also be used for comparison of screening procedures in different substructure search systems.

REFERENCES AND NOTES

(1) Lancaster, F. W. "Information Retrieval Systems: Characteristics, Testing and Evaluation", 2nd ed.; Wiley-Interscience: New York, 1979.
(2) Salton, G.; McGill, M. J. "Introduction to Modern Information Retrieval"; McGraw-Hill: New York, 1983; Chapter 5.
(3) van Rijsbergen, C. J. "Information Retrieval", 2nd ed.; Butterworths: London, 1979; Chapter 7.
(4) Sparck Jones, K., Ed. "Information Retrieval Experiment"; Butterworths: London, 1981.
(5) Cleverdon, C. W.; Mills, J.; Keen, E. M. "Factors Determining the Performance of Indexing Systems"; College of Aeronautics: Cranfield, England, 1966.
(6) Lynch, M. F.; Harrison, J. M.; Town, W. G.; Ash, J. E. "Computer Handling of Chemical Structure Information"; MacDonald-Elsevier: New York, 1971; Chapter 6.
(7) Adamson, G. W.; et al. "Strategic Considerations in the Design of a Screening System for Substructure Searches of Chemical Structure Files". *J. Chem. Doc.* **1973**, *13*, 153-157.
(8) See, for instance, Adamson, G. W.; et al. "An Evaluation of a Substructure Search Screen System Based on Bond-Centred Fragments". *J. Chem. Doc.* **1974**, *14*, 44-48.

Combinatorial Problems in Computer-Assisted Structural Interpretation of Carbon-13 NMR Spectra

ALAN H. LIPKUS and MORTON E. MUNK*

Department of Chemistry, Arizona State University, Tempe, Arizona 85287

Received April 24, 1984

Combinatorial problems posed by a method for computer-assisted structural interpretation of ^{13}C NMR spectra based upon fragments consisting of a carbon atom and its α neighbors are discussed. The basic problem of generating all structures consistent with a set of inferred fragments that contains mutually exclusive alternatives is divided into two parts: generation of combinations of fragments and exhaustive assembly of each combination into molecules. Algorithmic solutions to both of these problems are presented in detail.

INTRODUCTION

The increasing importance of ^{13}C magnetic resonance spectroscopy as a tool in organic structure elucidation continues to stimulate the development of computer-based methods that aid the chemist in drawing structural inferences from the ^{13}C spectra of unknown compounds.[1] The most ambitious of these methods are those that automatically infer from the ^{13}C spectrum a set of possible substructures and then generate all candidates for the unknown that are consistent with these substructures as well as with the molecular formula and, if desired, information derived from other sources, spectroscopic or chemical. The basic strategy used is one of substructure inference followed by structure generation.

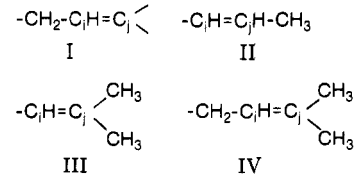
This strategy is part of a more general strategy employed by the chemist in the structure elucidation process. The computer modeling of this process is the goal of the evolving CASE system of programs.[2] The three major components of the system are spectrum interpretation (INTERPRET), molecule assembly (ASSEMBLE), and spectrum simulation and comparison (SIMULATE). The search for a convenient and efficient link in CASE between programs INTERPRET and ASSEMBLE has motivated the present work in combinatorial computing.

In general, computer systems for the structural interpretation of ^{13}C NMR spectra must use combinatorial algorithms capable of generating molecular structures from substructures that may overlap to some extent. Also, given that present-day knowledge of the relationships between molecular structure and spectral properties is incomplete and subject to inherent limitations, two or more substructures can arise from alternative interpretations of the same spectral feature. Thus, the algorithms used must be able to generate molecular structures from a set of substructures that may include mutually exclusive alternatives. Previous approaches have used different algorithms as well as different substructures.

In the CHEMICS system, designed by Sasaki et al.,[3] the selection of possible substructures based upon the ^{13}C spectrum of an unknown is accomplished by deleting from a list of small substructures (called components) all those that are shown by a correlation table of components and chemical shift ranges to be inconsistent with the spectrum. (The program can also use ^{1}H NMR, IR, and MS data to delete components.) From the set of remaining components, all subsets are formed that are consistent with the molecular formula and account for all ^{13}C resonances measured. From each subset of components, molecular structures are exhaustively generated by an appropriate algorithm,[4] thus producing all plausible candidates for the unknown compound. In later versions,[5] structure generation can be constrained by requiring the presence or absence of specific substructures.

Gray et al.[6] have developed a ^{13}C NMR interpretation method that uses a data base of substructures derived from a library of assigned spectra. Each substructure describes the environment of a particular carbon nucleus and is matched to the chemical shift and multiplicity of that nucleus. For each ^{13}C resonance in the spectrum of an unknown compound, those substructures that display the same multiplicity and a similar shift are selected from the data base. A reduction in the number of retrieved substructures is then attempted by an

0095-2338/85/1625-0038$01.50/0



I II III IV

Figure 1. Substructures illustrating the assembly process of Dubois et al.[9]

iterative process that tests whether any substructure is inconsistent with the others; if so, it is eliminated. The remaining substructures, some of which may result from alternative interpretations of the same ^{13}C resonance, are then passed to the structure generation program GENOA.[7] The chemist can submit constraints inferred from other data to this program so as to restrict the number of structures it produces.

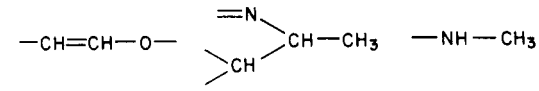
In this paper, we describe algorithmic solutions to some combinatorial problems suggested by another method for ^{13}C NMR spectrum interpretation. The basic unit of substructure it employs consists of a carbon atom and all of its α neighbors. One possible scheme for spectrum interpretation using this substructural unit has been briefly described by Dubois et al.[1,8,9] A data base of these substructures, derived from a library of assigned ^{13}C NMR spectra, is indexed so that a substructure can be retrieved through any of the ordered pairs composed of the chemical shift of the central atom and the shift of any one of its neighboring carbon atoms. Given the ^{13}C spectrum of an unknown compound, all possible ordered pairs of the given chemical shifts are formed. The data-base index is searched for similar pairs through which substructures are retrieved, thus creating a set of possible substructures of the unknown.

A procedure for assembling molecules from these substructures has been outlined by Dubois et al.;[9] it uses the presence or absence of one-bond overlaps between substructures to guide the assembly process. As an illustration, assume that substructure I in Figure 1 has been selected as an interpretation of the chemical shift pair (δ_i, δ_j) and that substructures II and III are alternative interpretations of (δ_j, δ_i) (note that signal multiplicity information has not been used). Because of the mismatch between the carbon atoms labeled j in substructures I and II, these substructures cannot be overlapped. On the other hand, substructures I and III can be assembled into IV. To continue the assembly process, intermediate structures such as IV are tested for overlap with other substructures. (A procedure for building a molecule whose carbon connectivity is broken by heteroatoms, e.g., ether oxygens, was not reported.)

Generally, to produce all candidates for an unknown from a set of possible substructures that may contain mutually exclusive alternatives, one must (i) form subsets of mutually nonexclusive substructures and (ii) exhaustively assemble each subset into molecules. These two tasks are performed separately in the CHEMICS system, but they are not explicitly separated in the method of Dubois et al. Because the former approach appears to us to be more easily and efficiently implemented, we have developed algorithms that perform consecutively the combinatorial tasks i and ii using the same substructural unit used by Dubois et al. To do this, we have posed the basic problem differently.

FORMULATION OF THE PROBLEM

A convenient way to describe the non-hydrogen atoms in a molecule is with an "atom type" that includes the element, its hybridization, which may additionally specify aromaticity, and the number of attached hydrogens. The carbon types not involving aromatic bonds, for example, are $-CH_3$, $=CH_2$, $-CH_2-$, $\equiv CH$, $=CH-$, $>CH-$, $\equiv C-$, $=C=$, $>C=$, and $>C<$. The substructural unit to be used (henceforth called a fragment) can now be more accurately defined as a carbon type and all of its nearest-neighbor atom types. Examples of fragments are



In the algorithm development to follow, it is assumed that the molecular formula of the unknown compound can be partitioned into a set of atom types. This would require more data than is used by Dubois et al., but the supplementation of ^{13}C chemical shift information with signal multiplicities often allows a complete determination of the carbon types in an unknown. Determination of the heteroatom types may require the application of several spectroscopies. If the available data do not make possible a unique molecular formula partition, then each plausible partition forms the basis of a separate interpretation problem, and the resulting structures can be pooled to form an irredundant set.

It is further assumed that each signal in the ^{13}C NMR spectrum has assigned to it one carbon type taken from the partition and that every carbon type in the partition is so assigned. In performing this assignment, any signal arising from magnetically equivalent nuclei must be identified and is treated as a set of signals—one for each isochronous nucleus—having the same chemical shift; thus, the number of signals is made equal to the number of carbons. If more than one plausible assignment can be made, separate interpretation problems may be posed.

The desired manner of organizing the possible fragments, which is based upon this spectrum assignment, is to associate with each resonance in the ^{13}C spectrum a list of all fragments that represent possible local environments of the resonating nucleus. More precisely, let $[F_1(\delta_i), F_2(\delta_i), ..., F_n(\delta_i)]$ be the list of n fragments that arise from alternative structural interpretations of the ith signal, which has shift δ_i (dependence upon signal multiplicity is implied). All of the fragments in this list have at their center the same carbon type, viz., the one assigned to the ith ^{13}C signal. The overall organization of possible fragments is a set of lists:

$$\{[F_1(\delta_1), F_2(\delta_1), ..., F_{n(1)}(\delta_1)]; [F_1(\delta_2), F_2(\delta_2), ..., F_{n(2)}(\delta_2)]; ...; [F_1(\delta_N), F_2(\delta_N), ..., F_{n(N)}(\delta_N)]\}$$

where N is the number of signals (carbons) and $n(i)$ is the number of alternative fragments associated with the ith signal.

The two combinatorial tasks to be performed can now be restated. First, generate all combinations of fragments of the form

$$[F_{s(1)}(\delta_1), F_{s(2)}(\delta_2), ..., F_{s(N)}(\delta_N)], \qquad 1 \leq s(i) \leq n(i)$$

These combinations are produced by selecting one fragment from each list. Second, use one-bond overlaps to exhaustively assemble each combination of fragments, along with all the heteroatom types in the partition, into molecules. Because the number of combinations of fragments may be prohibitively large, the first task requires the application of tests that discourage the formation of combinations that cannot be assembled into molecules.

GENERATION OF COMBINATIONS OF FRAGMENTS

A useful graphical representation of many combinatorial problems involving exhaustive enumeration is a search tree that can be traversed in such a way as to yield all solutions.[10] The top of a search tree for the generation of combinations of fragments is depicted in Figure 2. Starting with level 0, which contains only the root node, every node at level l is connected by downward emanating edges to a set of $n(l + 1)$ descendants

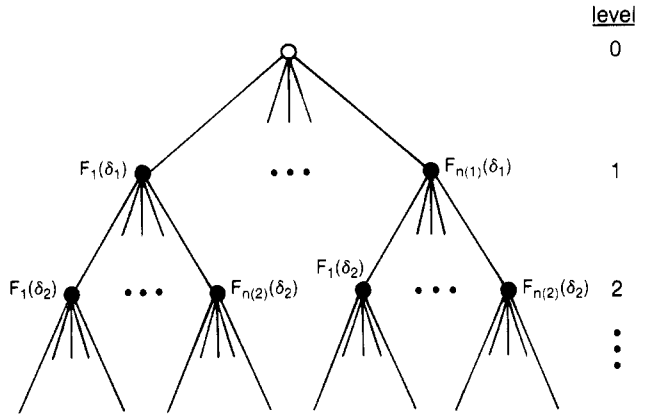


Figure 2. Search tree for the generation of combinations of fragments. $F_i(\delta_j)$ is the ith fragment in the list of alternatives associated with the jth ^{13}C signal.

at level $l + 1$. Each of these descendants represents one fragment from the list of alternatives for the $(l + 1)$th ^{13}C signal; the representation is always in the same left-to-right order: $F_1(\delta_{l+1})$, $F_2(\delta_{l+1})$, ..., $F_{n(l+1)}(\delta_{l+1})$. If there are N signals (carbons) in the interpretation problem, the nodes at level N have no descendants and are called terminal nodes.

For our purpose, the most convenient way to traverse this tree is in depth-first order, which proceeds iteratively as follows: from the current node, move downward to the leftmost node among its descendants that has not yet been visited; if all descendants have been visited or the current node is terminal, then "backtrack", i.e., move upward to the most recently visited node at the previous level. The search begins with the root as current node. When backtracking from the root is attempted, the search ends.

All combinations of fragments can be generated if a depth-first search of the tree in Figure 2 is used to guide the construction of a "search vector" according to the following rules: when a new node is visited by descending an edge, "expand" the vector by adding to it the fragment associated with that node; when backtracking, delete from the vector the most recently added fragment. The search vector is empty at the root. It represents one of the desired combinations (a terminal combination) whenever a terminal node is reached. At a nonterminal node, the search vector represents a partial combination of fragments.

As previously noted, the number of combinations may be so large that their generation would be prohibitively time consuming. In order to avoid this, expansions of partial combinations are subjected to two tests. A consistency test discourages the generation of combinations that cannot be assembled into molecules by checking for consistency between the molecular formula partition and certain inferences drawn from the partial combination. A redundancy test detects the formation of identical combinations of fragments via different paths in the search tree. If the latest expansion fails either test, it is rejected, and backtracking from the current node takes place. The efficiency advantage of such backtracking is great because it eliminates the need to search the potentially large subtree that has the current node as its root.

Consistency Test. It can be assumed that the simplest sort of consistency with the molecular formula partition has already been enforced by disallowing any fragment that contains an atom type not in the partition or contains an atom type more times than it appears in the partition. To illustrate the nature of the test (actually, a set of tests) to be derived, consider the expansion of the partial combination

$$C_1 = [-CH_2-CH{=}CH-,\ -CH_2-C{\equiv}C-]$$

into

$$C_2 = [-CH_2-CH{=}CH-,\ -CH_2-C{\equiv}C-,\ {=}CH-CH_2-C{\equiv}]$$

or

$$C_3 = [-CH_2-CH{=}CH-,\ -CH_2-C{\equiv}C-,\ {=}CH-CH_2-CH{=}]$$

when the molecular formula partition contains one methylene carbon. The expansion $C_1 \rightarrow C_2$ is consistent with this partition, but $C_1 \rightarrow C_3$ is inconsistent because it can be inferred from C_3 that there must be two different methylene carbons present. Such inferences depend upon the recognition that the same carbon atom can appear several times in a combination: the central carbon in fragment F can appear as a neighbor in fragment F′ when the center of F′ is a neighboring carbon in F. A simple test for this reciprocal relationship between adjacent carbons in different fragments was used by Beech et al.[11] in their computer program for the structural interpretation of proton NMR spectra. This relationship will presently be used to develop a more quantitative test.

Consider a combination of fragments in terms of its constituent bonds, preserving the distinction between central and neighboring atom types. Let b^m_{CA} be the total number of bonds of a given multiplicity between a central carbon type denoted by C and a neighboring atom type denoted by A. The superscript denotes the multiplicity of the bond: $m = 1$ for single bond, $m = 2$ for double, or $m = 3$ for triple (an aromatic bond could also be denoted). As an example, assume that subscript 0 denotes the carbon type —C≡, 1 denotes —CH═, and 2 denotes —CH_2—. The partial combination C_3 (see above) is then characterized by the quantities

$$b^1_{00} = b^1_{01} = b^1_{10} = b^1_{11} = b^1_{20} = b^1_{22} = 0$$

$$b^3_{00} = b^1_{02} = b^2_{11} = b^1_{12} = 1$$

and

$$b^1_{21} = 2$$

A necessary condition for a terminal combination of fragments to be capable of assembly into one or more molecules is that

$$b^m_{CC'} = \begin{cases} b^m_{C'C} & \text{if } C \neq C' \\ b^m_{CC} + (b^m_{CC} \bmod 2) & \text{if } C = C' \end{cases} \quad (1)$$

where C and C' denote any carbon types whose hybridizations permit mutual bonding and m can assume any compatible value. This is the aforementioned reciprocal relationship in quantitative terms. It reflects the counting twice of every bond between carbons: once when either carbon is at the center of a fragment. The "mod" operation[12] (I mod J equals the remainder when I is divided by J) has been used to formalize the requirement that b^m_{CC} be even. It is the implications of eq 1 for testing partial combinations that will be developed.

Define d^m_A as the number of bonds of multiplicity m in which atom type A participates in a molecule. For example, if A denotes the carbon type >C═, then $d^1_A = 2$, $d^2_A = 1$, and $d^3_A = 0$. For any combination of fragments, if p_C is the number of times that carbon type C appears in the combination as the center of a fragment, then

$$p_C = (\sum_c b^m_{Cc} + \sum_h b^m_{Ch})/d^m_C \quad (2)$$

for all m such that $d^m_C > 0$. The first (second) summation is over all carbon types (heteroatom types) appearing in the molecular formula partition whose hybridizations are compatible with the particular value of m being used. (Throughout this paper the indexes of summation are written in lowercase.)

It must be assumed a priori that any partial combination under consideration can, by successive expansions, become a

terminal combination capable of assembly into one or more molecules. If it is expanded into such a combination, it will then satisfy eq 1 even if it presently does not. Thus, we assume that each $b^m_{CC'}$ eventually takes on a value, call it $B^m_{CC'}$, such that

$$B^m_{CC'} \geq b^m_{CC'} \tag{3}$$

and (cf. eq 1)

$$B^m_{CC'} = \begin{cases} B^m_{C'C} & \text{if } C \neq C' \\ B^m_{CC} + (B^m_{CC} \bmod 2) & \text{if } C = C' \end{cases} \tag{4}$$

For testing purposes, the most conservative estimate of $B^m_{CC'}$ is used. It is the smallest integer satisfying both eq 3 and 4, viz.

$$B^m_{CC'} = \begin{cases} \max (b^m_{CC'}, b^m_{C'C}) & \text{if } C \neq C' \\ b^m_{CC} + (b^m_{CC} \bmod 2) & \text{if } C = C' \end{cases} \tag{5}$$

where the "max" function equals the larger of its two arguments.

If b^m_{Cc} in eq 2 is replaced by B^m_{Cc}, then the right-hand side of that equation, rounded up to the next highest integer if necessary, represents the minimum number of carbon types C implied by the partial combination. This number cannot be greater than the total number of times, call it P_C, that carbon type C appears in the molecular formula partition. Thus, part of the consistency test is that

$$P_C \geq (\sum_c B^m_{Cc} + \sum_h b^m_{Ch})/d^m_C \tag{6}$$

for every C representing a carbon type appearing in the partition and all m such that $d^m_C > 0$. If a partial combination fails this test, backtracking takes place from the corresponding node in the search tree.

At a terminal node, the combination must instead obey strict equality in eq 6, i.e.

$$P_C = (\sum_c B^m_{Cc} + \sum_h b^m_{Ch})/d^m_C \tag{7}$$

It can be shown, as follows, that if a terminal combination satisfies eq 7, the reciprocity condition (eq 1) is also met. Any terminal combination satisfies (cf. eq 2)

$$P_C = (\sum_c b^m_{Cc} + \sum_h b^m_{Ch})/d^m_C \tag{8}$$

Subtraction of eq 8 from eq 7 yields

$$\sum_c (B^m_{Cc} - b^m_{Cc}) = 0$$

Since the term in parentheses can never be negative, each such term in the summation must equal 0. It then follows from eq 4 that eq 1 is satisfied. Conversely, if the reciprocity condition is satisfied, eq 7 follows directly from eq 8 by substitution.

The other part of the consistency test, based upon heteroatom types appearing in the molecular formula partition, cannot be developed in the previous manner because heteroatoms serve only as neighbors, not centers, in fragments. Let P_H be the number of times that heteroatom type H appears in the partition. Any combination, partial or terminal, must satisfy the inequality

$$P_H \geq \sum_c b^m_{cH}/d^m_H \tag{9}$$

for every H representing a heteroatom type in the partition and all m such that $d^m_H > 0$. Equality in eq 9 cannot be required because of the possibility of heteroatom–heteroatom bonding.

A terminal combination that passes all relevant parts of the consistency test may still be incapable of assembly into molecules because the reciprocity condition (eq 1) is necessary but not sufficient. It does not recognize those features that distinguish a chemical graph, i.e., a molecular structure, from graphs in general:[13] connectedness (note that a combination capable only of assembly into two separate molecules also satisfies eq 1), absence of multiple edges (this assumes that a multiple bond is not represented by a multiple edge but by an edge of a particular "color"), and absence of loops [a loop is an edge that joins a vertex (atom) to itself]. But these constraints are imposed upon the structure generation process.

Redundancy Test. Up to now, we have treated a search vector the same as a combination of fragments, but there is one difference: fragments in the search vector are explicitly ordered. Because of this, it is possible for vectors associated with different nodes in the search tree to correspond to the same combination, differing only in their ordering of fragments. All of these vectors except the one generated first in the tree search can be considered redundant and should not be expanded further. The reason for this is that searching any of the subtrees having one of these nodes as its root will result in the same terminal combinations because the consistency test is independent of fragment ordering.

This redundancy exists when two or more lists of alternative fragments have fragments in common—probably a frequent occurrence in ^{13}C spectral interpretation. For example, assume that the search vector

$$V = [F_a(\delta_1), \ldots, F_i(\delta_I), \ldots, F_j(\delta_J)]$$

is associated with the current node in the tree. Assume that fragments $F_i(\delta_I)$ and $F_{j'}(\delta_J)$ are identical and that $F_{i'}(\delta_I)$ and $F_j(\delta_J)$ are also identical. V thus corresponds to the same combination of fragments as

$$V' = [F_a(\delta_1), \ldots, F_{i'}(\delta_I), \ldots, F_{j'}(\delta_J)]$$

In this example, if $i' < i$, then V is redundant. The computing time wasted on redundant searching, if it is not avoided by backtracking, can become disproportionately large as the extent to which lists have fragments in common increases.

One way of determining redundancy is to compare newly generated search vectors with the contents of a file containing all previously generated vectors that were shown by a similar comparison to be irredundant. This requires the storage of what may become a very large file. A method that does not require increased storage is based upon tree searching.

Let V_0 be the search vector associated with the current node, residing at level l, in the tree that generates combinations of fragments (Figure 2), and let V_1 be a new vector to be constructed via a depth-first search of the same tree. In this new search, which begins at the root node with V_1 empty, the latest expansion of V_1 must satisfy the following two tests or else be rejected by backtracking: (1) No fragment may appear in V_1 more times than it appears in V_0; (2) the number of fragments in V_1 is either (a) less than l or (b) equal to l. The search ends the first time that V_1 satisfies both tests 1 and 2b. Clearly, V_1 is then either a permutation of V_0 or is equal to V_0; if V_1 is a permutation, it is the one generated first in the tree search. Thus, V_0 is redundant if and only if $V_1 \neq V_0$.

The tree shown in Figure 2 is constructed on the basis of an arbitrary numbering (1, 2, ..., N) of the ^{13}C signals. It is easy to rearrange the tree by simply changing this numbering. Some arrangements may be far more efficient than others in that they require a much smaller portion of the tree to be traversed. A rule that often leads to the most efficient tree is to put those nodes having the fewest descendants nearest to the top.[10] This suggests that the ^{13}C signals should be numbered so that $n(1) \leq n(2) \leq \ldots \leq n(N)$, thereby putting the smallest lists nearest to the top.

Table I. A Set of Lists of Fragments to Illustrate the Generation of Combinations of Fragments

list no.[a]	fragments[b]
1	$CH_2(=C<)$[c,d]
2	$CH_3(-CH<)$, $CH_3(>C<)$[c,d]
3	$CH_3(-CH<)$, $CH_3(>C<)$[c,d]
4	$CH_3(-CH_2-)$,[c] $CH_3(>C<)$[d]
5	$CH_2(-CH_2-)(-CH<)$,[c,d] $CH_2(-CH_2-)(>C<)$
6	$CH_2(-CH_2-)(-CH<)$,[d] $CH_2(-CH_2-)(>C<)$[c]
7	$CH_2(-CH_3)(-CH_2-)$, $CH_2(-CH_2-)(-CH_2-)$,[d] $CH_2(-CH_2-)(-CH<)$[c]
8	$CH_2(-CH_3)(>C=)$,[c] $CH_2(-CH_2-)(-CH<)$, $CH_2(-CH_2-)(>C<)$[d]
9	$CH_2(-CH_3)(>C=)$, $CH_2(-CH_2-)(-CH<)$, $CH_2(-CH_2-)(>C<)$[c,d]
10	$C(-CH_2-)(-CH<)(=CH_2)$,[c] $C(-CH_2-)(>C<)(=CH_2)$, $C(-CH<)(>C<)(=CH_2)$,[d] $C(>C<)(>C<)(=CH_2)$
11	$CH(-CH_3)(-CH<)(>C=)$, $CH(-CH_3)(>C<)(>C=)$, $CH(-CH_2-)(-CH<)(>C=)$,[d] $CH(-CH_2-)(>C<)(>C=)$[c]
12	$CH(-CH_2-)(-CH_2-)(-CH<)$, $CH(-CH_2-)(-CH_2-)(>C<)$, $CH(-CH<)(-CH<)(>C<)$,[c,d] $CH(-CH<)(-CH<)(>C=)$
13	$CH(-CH_3)(>C<)(>C<)$, $CH(-CH_2-)(-CH<)(>C<)$,[d] $CH(-CH_2-)(-CH<)(>C=)$, $CH(-CH_2-)(>C<)(>C<)$,[c] $CH(-CH<)(-CH<)(>C=)$
14	$C(-CH_3)(-CH_3)(-CH_2-)(-CH_2-)$, $C(-CH_3)(-CH_3)(-CH_2-)(-CH<)$,[d] $C(-CH_3)(-CH_3)(-CH_2-)(>C<)$, $C(-CH_3)(-CH_3)(-CH<)(>C=)$, $C(-CH_3)(-CH_2-)(-CH<)(-CH<)$[c]
15	$C(-CH_3)(-CH_3)(>C<)(>C=)$, $C(-CH_3)(-CH_2-)(-CH<)(-CH<)$,[c] $C(-CH_3)(-CH_2-)(-CH<)(>C<)$, $C(-CH_3)(-CH_2-)(-CH<)(>C=)$,[d] $C(-CH_3)(-CH_2-)(>C<)(>C=)$

[a] One list of fragments is given for each carbon type in a partition of the molecular formula $C_{15}H_{24}$. [b] In the linear notation used, the central carbon type is first, followed by its neighboring carbon types in parentheses. [c] Fragment selected for valid combination 1. [d] Fragment selected for valid combination 2.

In the type of tree being considered, other factors may weaken this rule. Because of the positive correlation expected between the number of fragments in a list of alternatives and the number of atom-type neighbors within each fragment, fragments containing the largest numbers of neighbors would likely be placed near the bottom of the tree. But since the consistency test becomes more severe as more bonds are accumulated, it might be better to encounter fragments having many neighbors earlier in the search. Also, it is desirable to arrange the tree so that redundant search vectors are encountered as early as possible. In practice, however, a tree arranged according to the rule stated above should be adequately efficient.

The great efficiency advantage that can be realized by using the consistency and redundancy tests when generating combinations of fragments is illustrated by the simple hypothetical problem described in Table I. A search tree for this problem, constructed in the manner depicted in Figure 2, would have 8 644 767 nodes. In the actual search, however, only 2222 nodes were visited. Among these nodes, the consistency test was failed at 1519 of them, and the redundancy test was failed (redundancy detected) at another 154. As indicated in Table I, the search found only two valid combinations, i.e., terminal combinations that passed the consistency and redundancy tests.

GENERATION OF STRUCTURES

The immediate goal of the interpretation process is the generation of all plausible candidates for the unknown compound. Every valid combination of fragments must be exhaustively assembled, along with all heteroatom types in the molecular formula partition, into molecules. This can be accomplished with the aid of any structure-generation program that can accept overlapping fragments as input. Every time a valid combination is produced, the structure generator is called. When the execution of this program is complete, the tree search for more valid combinations is resumed, starting by backtracking from the current terminal node.

The program ASSEMBLE, one component of the CASE system, can be used as the structure generator since it successfully treats one-bond overlaps between substructures.[2] But a more specialized structure generator is possible and will be developed in this section. Since it is designed specifically to handle fragments, it is faster. Efficiency is important because, as noted earlier, not every valid combination can be assembled into molecules. The time-consuming effect of this is offset by efficient structure generation.

A search tree is again useful as a graphical representation. The generation of structures can be accomplished by depth-first searching of a tree each of whose nodes, except the root, represents a specific atom (type)–atom (type) connection. The tree search guides the construction of a search vector by the same method described in the previous section. This vector is empty at the root, but at every other node in the tree it contains atom–atom connections that together describe the connectivity of either a partially assembled molecule or, if a terminal node has been successfully reached, a fully assembled molecule.

Every atom–atom connection is denoted by an ordered pair of identification numbers. Each of these numbers (identifiers) represents uniquely a carbon or heteroatom type in the molecular formula partition. Although this numbering can be arbitrary, it is natural to derive it from the numbering of the N signals in the ^{13}C NMR spectrum; i.e., if $[F_{s(1)}(\delta_1), F_{s(2)}(\delta_2), ..., F_{s(N)}(\delta_N)]$ is a valid combination, the central carbon types in these fragments can be represented by the identifiers 1, 2, ..., N, respectively. This numbering can be extended ($N + 1$, $N + 2$, ...) arbitrarily to include all heteroatom types in the partition.

The search tree is built on the basis of the connectivity information contained in the fragments of the valid combination under consideration. (Thus, the tree is rebuilt every time a new valid combination is produced.) A result of using this information is that the search tree can be conveniently built out of subtrees, each one associated with a particular kind of bond (see Figure 3a). The root node of every subtree except the topmost is a terminal node of the previous subtree. The terminal nodes of the search tree are those of the bottommost subtrees.

Each subtree is used to perform the same kind of combinatorial task: exhaustively enumerating the ways of pairing all the elements of one set with those of another. Therefore, all of the subtrees have the same basic form. This is depicted in Figure 3b for the case in which the two sets are $\{e_1, e_2, ..., e_L\}$ and $\{e'_1, e'_2, ..., e'_L\}$. Every node at level l ($l < L$) has L descendants that represent, in left-to-right order, the pairs (e_{l+1}, e'_1), (e_{l+1}, e'_2), ..., (e_{l+1}, e'_L). [Using instead the ordered pairs (e'_{l+1}, e_1), (e'_{l+1}, e_2), ..., (e'_{l+1}, e_L) leads to an equivalently functioning subtree.] The nodes at level L are terminal nodes.

The particular kind of bond with which each subtree is associated depends upon the interpretation given to the two sets whose pairings are to be enumerated. With one exception, the elements of these sets are always atom-type identifiers. Each ordered pair represents an atom–atom connection of specific bond multiplicity. All subtrees necessary to build the search tree fall into one of three categories: carbon–carbon, carbon–heteroatom, and heteroatom–heteroatom.

Carbon–Carbon Subtrees. From this category, there will be one subtree for each nonzero value of $b^m_{CC'}$ where C and

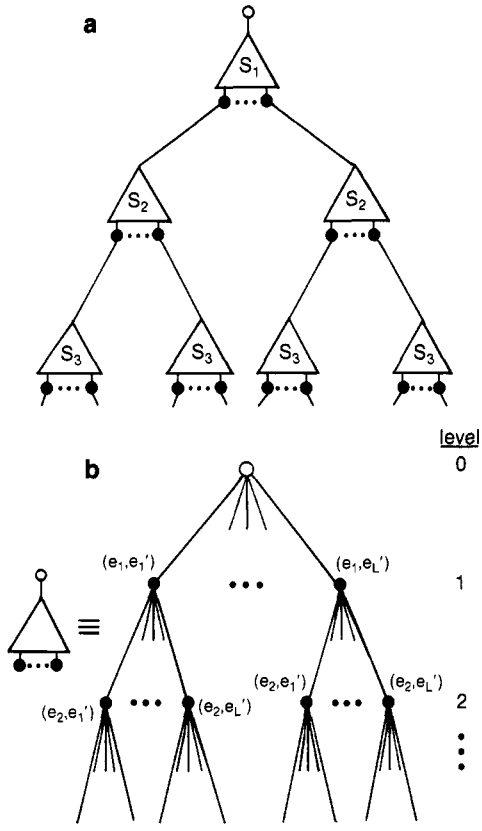


Figure 3. Search tree for the generation of structures: (a) representation of the tree in terms of subtrees; (b) typical subtree for enumerating the ways of pairing all the elements of two sets—in this case $\{e_1, e_2, ..., e_L\}$ and $\{e'_1, e'_2, ..., e'_L\}$.

C' ($C \geq C'$) denote carbon types, characterizing the valid combination. The elements of one of the two sets are the identifiers of those carbon types C that participate in a bond of multiplicity m with a neighboring carbon type C'; the elements of the other set are the identifiers of carbon types C' that participate in a bond of multiplicity m with a neighboring carbon type C. The identifier of each carbon type appears in its respective set as many times as that carbon type participates in a CC' bond; consequently, there are $b^m_{CC'}$ elements in each set.

If $C = C'$, the two sets are identical. Since the b^m_{CC} elements that each set has is twice the number of CC bonds of multiplicity m, the following symmetry test must be enforced: a search vector that contains (e_i, e_j) must also (or eventually) contain the pair (e_j, e_i). The appearance of both of these ordered pairs in the search vector counts as one connection. (Note that if $b^m_{CC} = 2$, the sets $\{e_1, e_2\}$ and $\{e_1, e_2\}$ can be replaced by $\{e_1\}$ and $\{e_2\}$.) Another test, which must be applied to every subtree, is that a search vector containing (e_i, e'_k) must not contain any pair (e_j, e'_k) where $i \neq j$. This ensures that the elements of the two sets are paired in a one-to-one manner.

Carbon–Heteroatom Subtrees. This category contributes one subtree for every heteroatom type in the molecular formula partition and each compatible value of bond multiplicity. Assume that H denotes a heteroatom type and m denotes a multiplicity such that $d^m_H > 0$. The elements of one of the two sets are the identifiers of all heteroatom types H in the partition. These identifiers appear once for each possible bond of multiplicity m; thus, there are $P_H d^m_H$ elements in this set.

The elements of the other set are the identifiers of all carbon types that participate in a bond of multiplicity m with a neighboring heteroatom type H. The identifier of each carbon type appears once for each such bond that carbon type participates in; there are thus $\sum_c b^m_{cH}$ identifiers in this set. If the difference D^m_H, where

$$D^m_H = P_H d^m_H - \sum_c b^m_{cH}$$

is not zero, then the rest of this set consists of D^m_H "dummy" elements.

The quantity D^m_H can be interpreted as the total number, among all heteroatom types H, of bonding sites of multiplicity m that are engaged in bonding to other heteroatoms only. Any particular heteroatom type H must be engaged in bonding to other heteroatoms as many times as its identifier is paired with a dummy element in the search vector. This information is used to construct any heteroatom–heteroatom subtrees that are necessary.

Heteroatom–Heteroatom Subtrees. Each subtree in this category is used to form all heteroatom–heteroatom bonds of a particular multiplicity. There will be one subtree for each value of m such that $\sum_h D^m_h > 0$. There may, of course, be no subtrees from this category in a search tree even if there are heteroatoms in the problem.

The two sets to be paired are always identical. Each consists of the identifiers of all those heteroatom types that were paired with dummy elements in a carbon–heteroatom subtree associated with multiplicity m (the heteroatom–heteroatom subtrees are thus dependent upon the path taken through previous subtrees); the identifiers appear once for each such pairing. There are consequently $\sum_h D^m_h$ elements in a set. Since this is twice the number of heteroatom–heteroatom bonds of multiplicity m (see below), searching this subtree is subject to the same symmetry test described with regard to carbon–carbon subtrees.

To prove that $\sum_h D^m_h$ always is an even number for valid combinations, consider that any plausible partition of the molecular formula must satisfy the condition

$$(\sum_h P_h d^m_h + \sum_c P_c d^m_c) \bmod 2 = 0$$

for all m, where the first summation is over its heteroatom types and the second is over its carbon types. This implies that

$$(\sum_h P_h d^m_h - \sum_c P_c d^m_c) \bmod 2 = 0 \qquad (10)$$

But

$$\sum_c P_c d^m_c = \sum_c (b^m_{cc} + \sum_h b^m_{ch} + 2\sum_{c'>c} b^m_{cc'})$$

for valid combinations. Thus, after substitution, eq 10 becomes

$$(\sum_h P_h d^m_h - \sum_c \sum_h b^m_{ch} - I) \bmod 2 = 0$$

where I is some even number, or

$$(\sum_h D^m_h) \bmod 2 = 0$$

To ensure that the connections in the search vector describe the connectivity of a chemical graph whenever a terminal node in the search tree is reached, certain topological tests are applied to each newly expanded vector. These tests enforce those features, already mentioned, that distinguish chemical graphs. Connectedness is ensured by rejecting (via backtracking) the latest atom–atom connection if it forms a molecule not containing all atom types in the partition. Multiple edges and loops are prevented by rejecting connections that join two atoms previously connected or join an atom to itself.

Another test discourages redundant structure generation, i.e., the generation of molecules that differ only in the disposition of atom-type identifiers. The latest atom–atom connection is rejected if it is topologically equivalent to a connection in a previously searched branch of the tree, as determined from an algorithm that detects topological sym-

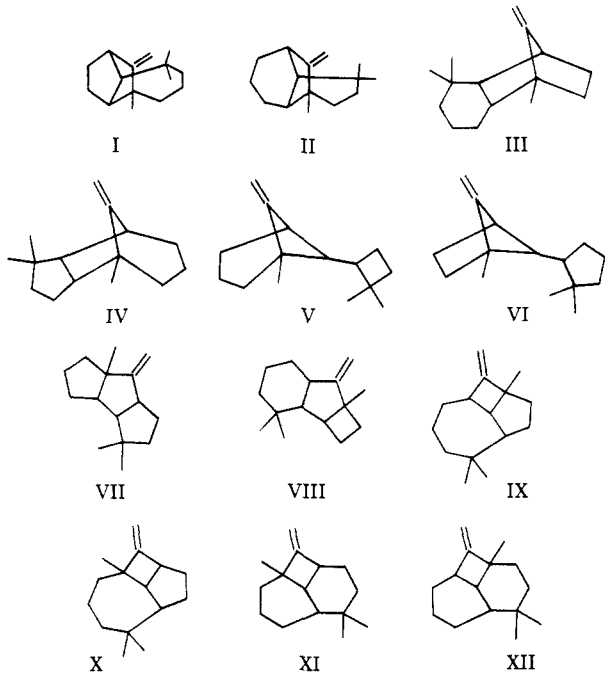


Figure 4. An exhaustive set of structures generated from valid combination 2 (see Table I). Structure I is the sesquiterpene longifolene.

metry in graphs.[14] Since prospective avoidance of redundant structures is not perfect, each newly generated structure must be assigned, via the same algorithm,[14] unique (canonical) atom identification numbers for comparison with previously generated structures. The comparison need be made only against those structures generated from the same combination of fragments because no molecule can be built from two different combinations of fragments. (If an aromatic bond is not used, however, different resonance structures of the same aromatic molecule may be built from different combinations.)

Because of the multiple appearances that subtrees make as "building blocks" of the search tree, the same subtree, in different locations, may be visited many times. However, a subtree is not necessarily traversed in the same way every time it is visited because the search will depend upon the path taken through previous subtrees.

Although the general scheme for linking subtrees to form the search tree has been described (Figure 3a), it has not been specified at which level a given subtree should appear. This order is arbitrary with one restriction: for each heteroatom type H and multiplicity m such that $D_H^m > 0$, the corresponding carbon–heteroatom subtree must be higher than the heteroatom–heteroatom subtree for multiplicity m.

The order of the subtrees will affect the efficiency of the search tree. The rule discussed in the previous section states that those nodes having the fewest descendants should be put nearest to the top. This rule can be implemented by ordering the subtrees according to L (see Figure 3b)—the lower the value of L, the higher the placement in the tree. If the above restriction is violated, the heteroatom–heteroatom subtree(s) can be moved downward.

The two valid combinations previously found in the problem described in Table I were used to obtain an example of output from this structure generator. No molecule was generated from combination 1, but a total of 12 molecules was generated from combination 2. These molecules are shown in Figure 4; they exemplify several different skeletal types. Of course, many combinations will lead to only one molecule. This number, however, may not always be relatively small. As a "worst" case, consider that the same combination of fragments that yields the chain molecule

$$CH(CH_3)_2CH(G_1)CH(G_2)...CH(G_n)CH(CH_3)(NH_2)$$

can be assembled into at least $n!$ different chain molecules if each branch (G_i) is unique.

CONCLUSIONS

The work in this paper was based on the assumption that the combinatorial problem of generating all structures consistent with a set of fragments that contains mutually exclusive alternatives could be greatly simplified by dividing it into two problems: the generation of all combinations of fragments and the exhaustive assembly of each combination into molecules. The relative simplicity of the algorithmic solutions that have been presented appears to justify this assumption.

From a practical standpoint, these solutions represent only the framework of a method since computer-assisted structure elucidation of typical unknown compounds is unlikely to be successful if only the ^{13}C NMR spectrum and molecular formula are used. Information derived from other spectroscopies must be used. The extension of the definition of a fragment to allow heteroatom types in the center may be useful in this regard.[15] Also, one can incorporate into the structure generator optional features (constraints) that can require the presence or absence of specific substructures or enforce various structural requirements not expressible in terms of substructures.

Powerful constraints already exist in program ASSEMBLE in the CASE system; they could be added to the specialized structure generator described in this paper. With a wide variety of constraints available, it is more likely that every piece of structural information that can be inferred from any spectroscopic or chemical data can be used to limit the number of generated structures.

ACKNOWLEDGMENT

We thank Prof. K. Balasubramanian for his helpful comments on the manuscript. Financial support by the National Institute of General Medical Sciences (NIH Grant GM 21703) and The Upjohn Co. is gratefully acknowledged.

REFERENCES AND NOTES

(1) For a comprehensive review, see Gray, N. A. B. "Computer Assisted Analysis of Carbon-13 NMR Spectral Data". *Prog. Nucl. Magn. Reson. Spectrosc.* **1982**, *15*, 201–248.
(2) Munk, M. E.; Shelley, C. A.; Woodruff, H. B.; Trulson, M. O. "Computer-Assisted Structure Elucidation". *Fresenius' Z. Anal. Chem.* **1982**, *313*, 473–479.
(3) Yamasaki, T.; Abe, H.; Kudo, Y.; Sasaki, S. "CHEMICS: A Computer Program System for Structure Elucidation of Organic Compounds". *ACS Symp. Ser.* **1977**, *No. 54*, 108–125.
(4) Kudo, Y.; Sasaki, S. "Principle for Exhaustive Enumeration of Unique Structures Consistent with Structural Information". *J. Chem. Inf. Comput. Sci.* **1976**, *16*, 43–49.
(5) Sasaki, S.; Fujiwara, I.; Abe, H.; Yamasaki, T. "A Computer Program System – NEW CHEMICS – for Structure Elucidation of Organic Compounds by Spectral and Other Structural Information". *Anal. Chim. Acta* **1980**, *122*, 87–94. Oshima, T.; Ishida, Y.; Saito, K.; Sasaki, S. "CHEMICS-UBE, A Modified System of CHEMICS". *Anal. Chim. Acta* **1980**, *122*, 95–102.
(6) Gray, N. A. B.; Crandell, C. W.; Nourse, J. G.; Smith, D. H.; Dageforde, M. L.; Djerassi, C. "Computer-Assisted Structural Interpretation of Carbon-13 Spectral Data". *J. Org. Chem.* **1981**, *46*, 703–715.
(7) Carhart, R. E.; Smith, D. H.; Gray, N. A. B.; Nourse, J. G.; Djerassi, C. "GENOA: A Computer Program for Structure Elucidation Utilizing Overlapping and Alternative Substructures". *J. Org. Chem.* **1981**, *46*, 1708–1718.
(8) Dubois, J. E.; Carabedian, M.; Ancian, B. "Élucidation Structurale Automatique par RMN du Carbone-13: Methode DARC-EPIOS. Recherche d'une Relation Discriminante Structure-Déplacement Chimique". *C. R. Hebd. Seances Acad. Sci., Ser. C* **1980**, *290*, 369–372.
(9) Dubois, J. E.; Carabedian, M.; Ancian, B. "Élucidation Structurale Automatique par RMN du Carbone-13: Methode DARC-EPIOS. Description de l'Élucidation Progressive par Intersection Ordonnée de Sous-Structures". *C. R. Hebd. Seances Acad. Sci., Ser. C* **1980**, *290*, 383–386.

(10) Reingold, E. M.; Nievergelt, J.; Deo, N. "Combinatorial Algorithms: Theory and Practice"; Prentice-Hall: Englewood Cliffs, NJ, 1977; Chapter 4.
(11) Beech, G.; Jones, R. T.; Miller, K. "Structural Interpretation of Proton Magnetic Resonance Spectra by Computer: First-Order Spectra". *Anal. Chem.* **1974**, *46*, 714-718.
(12) Knuth, D. E. "The Art of Computer Programming"; Addison-Wesley: Reading, MA, 1968; Vol. 1, Chapter 1, p. 38.
(13) Harary, F. "Graph Theory"; Addison-Wesley: Reading, MA, 1969; Chapter 2.
(14) Shelley, C. A.; Munk, M. E. "An Approach to the Assignment of Canonical Connection Tables and Topological Symmetry Perception". *J. Chem. Inf. Comput. Sci.* **1979**, *19*, 247-250.
(15) Farkas, M.; Munk, M. E., work in progress.

Monte Carlo Studies of the Classifications Made by Nonparametric Linear Discriminant Functions

TERRY R. STOUCH and PETER C. JURS*

The Pennsylvania State University, University Park, Pennsylvania 16802

Received August 28, 1984

Chance factors in pattern recognition studies utilizing nonparametric linear discriminant functions are examined. The relationship between complete linear separation of a data set and the dimensionality of the study is well-known. Also, due to the nature of the inequalities from which these numerical techniques are derived, 50% separation is always assured. This paper investigates the probability of achieving less than 100% but greater than 50% chance separations as a function of the dimensionality and class membership distribution. It is shown that the fraction of correct classifications due to chance factors increases dramatically as the dimensionality of the study increases. These results serve to redefine the level of expected chance classifications as a function of the number of observations, the dimensionality, and the class membership distributions. The results can be used to assess the classification results obtained with a given linear discriminant function.

The field of pattern recognition (PR) consists of techniques that are designed to classify numerical patterns.[1,2] When PR is applied to structure–activity relationship (SAR) studies, the patterns consist of measures of the compounds' physical and structural properties, and the classes are composed of compounds of like activity. These methods are useful when the activities of compounds are not known quantitatively. In such a case, the compounds can be assigned to a class of activity such as active, inactive, or very active. Such situations often occur when quantitative measures of activity are difficult, impractical, or impossible to obtain.

Discriminant generating methods of PR develop boundaries between the classes. The boundaries defined by linear discriminant functions (LDFs) are linear combinations of the variables and can be thought of as their corresponding geometrical forms: a line in two dimensions, a plane in three dimensions, and a hyperplane in higher dimensions. Algebraically, they are represented by eq 1, where the coefficients

$$f(\text{activity}) = a_0 + a_1x_1 + a_2x_2 + \ldots + a_dx_d \qquad (1)$$

of the variables (a_i's) comprise the discriminant and the variables (x_i's) are the measurements (descriptors) that describe the compounds and that comprise the patterns. d is the number of descriptors used in the discriminant. Parametric methods of discriminant generation use class statistics in order to generate discriminants; to do this, assumptions must be made concerning the distribution of the data. Nonparametric methods make no such assumptions and generate discriminants by analyzing each individual pattern. For this reason, many SAR PR studies have been performed with the nonparametric techniques.

It is well-known that, for a given number of compounds, the probability of fortuitously obtaining 100% complete and correct classification of the compounds increases as the number of features increases from 1 to the number of patterns in the study, N. This probability can be calculated with eq 2, where

$$P = 2\sum_{i=0}^{d} C_i^{N-1}/2^N \qquad (2)$$

$C_i^{N-1} = (N-1)!/[(N-1-i)!i!]$. N is the number of compounds, and d is the dimensionality, the number of descriptors. Figure 1 shows a plot of this relationship for 50 compounds. The only assumption that is made concerning the data is that it be in general position, that is, that no $d+1$ data points should be contained in a $d-1$ hyperplane. When a small number of descriptors is used to develop the LDF, the probability of achieving complete separation due to chance is small. As the number of descriptors approaches the number of compounds used in the study, however, the probability of such an occurrence increases. These classifications, while correct, are due only to artifacts of the mathematics governing the LDF-generating process. They are not due to any relationship between the compounds, and the resulting LDF will have no predictive ability beyond random guessing. This relationship has been known for some time and can be found in the literature.[1,2] Stuper and Jurs have shown that if the number of descriptors is kept below one-third the number of compounds used, the probability of complete separation due to chance can be kept low.[3] At the other extreme, it is intuitively obvious that for classes of equal size, 50% separation can be achieved by assigning all of the patterns in the study to one class. Often, however, LDFs are generated and used without reference to the levels of classification between these extremes. Often, random results are assigned the value of 50% correctly classified, and any result greater than 50% is considered to be due to the information contained within the descriptors and the explanation of at least a portion of the underlying SAR. This is contrary to the fact that the probability of 100% separation increases with an increasing number of descriptors. If the probability of 100% separation increases, then so too must the probability of less complete separations. This will have the effect of changing the level of classification that is considered

0095-2338/85/1625-0045$01.50/0