Molecular Structure Comparison Program for the Identification of Maximal Common Substructures[1]

Michael M. Cone, Rengachari Venkataraghavan, and Fred W. McLafferty*

Contribution from the Spencer T. Olin Laboratory, Department of Chemistry, Cornell University, Ithaca, New York 14853. *Received April 12, 1977*

Abstract: An algorithm has been developed which compares two molecules to identify the maximal substructures which are in common. Dot-plot symbolism based on Wiswesser line notations is used to generate binary occurrence vectors representing the molecules. A compatibility table is constructed from these vectors; "k-cover" incompatibility positions are cleared as well as positions not in agreement with "neighbor lists" and "degree lists" representing the connectivities of the molecules. A special analysis of this optimized compatibility table then yields the longest node-string paths, which represent the optimal substructures. Computational time requirements vary from 0.2 s for simple aliphatic and monocyclic compounds to 100 s for steroids.

Computer-aided examination and manipulation of chemical structure information has proved to be of significant value in a variety of applications. The comparison of structural groupings has attracted substantial attention, in particular for the "subgraph isomorphism problem" in which a particular molecular structure is inspected by the algorithm for the presence of the desired substructural unit, or for the comparison of two complete molecular structures for identity.[2-8] For the problem of finding the maximal molecular fragments (substructures) which are common to two molecules, one solution has been proposed by Lynch and co-workers,[9] describing computer programs for acyclic structures. However, for the organization and retrieval of molecular reaction data "this approach was abandoned because the complexity of the programs was uneconomic".[10] The problem of finding the maximal common substructures is an example of the derivation of the maximal common subgraphs of two undirected labeled graphs, and may be approached by either the exhaustive iterative node-by-node comparison of derived subsets (used by Lynch and co-workers[9]) or by the analysis of a "compatibility table".[5-7] As shown by Levi,[7] the latter should be more efficient as it first establishes the compatibilities between the nodes of the two graphs (vide infra).

Our particular need for such an algorithm arose in connection with the application of the "self-training interpretive and retrieval system" (STIRS) for unknown mass spectra, which for particular "data classes" produces lists of reference molecules which should contain various substructural features of the unknown molecule.[11] Because the present STIRS system contains 15 data classes, an algorithm that for each of the resulting lists could cross-compare all pairs of reference molecules to find the maximal substructures of highest frequency would be of substantial aid to the interpreter.

Other useful applications of such an algorithm can be envisioned. For the computer-assisted design of complex organic syntheses,[12] cross-comparison of the molecular structure of the compound sought against a list of available starting materials could provide an efficient identification of potential molecular building blocks. A wide variety of structure/activity studies can be envisioned; for the design and synthesis of new drugs of specific pharmacological activity, a computer cross-comparison of the molecular structures of known such drugs could provide an exhaustive list of the possible active molecular fragments.[13] For other spectroscopic methods of structure determination such an algorithm could examine molecules with spectral characteristics similar to that of the unknown, analogous to the application proposed here for unknown mass spectra.

Experimental Section

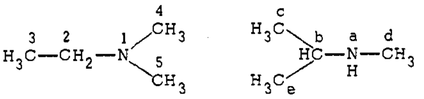
Our approach is based on the compatibility table method,[7] utilizing structural information from connection tables based upon dot-plot notation.[14] Wiswesser line notations were used to generate the dot-plot notation and connection tables, as the mass spectral reference file of 30 000 different compounds used for STIRS includes WLN designations. The generation program[15] is based on that developed for CROSSBOW.[14] Three symbols, those for spiro carbon, silicon, and SO_2 in a ring, were added to the original list of 23 symbols used to designate the nodes for the chemical structure. Note that these node symbols can refer to multiple atom and bonding combinations, such as a carbonyl or SO_2 group. The dot-plot notations for dimethylethylamine and methylisopropylamine are shown in Figure 1. Note that the algorithm is not limited to this representation of molecular structure; for connection table information the node values should be used.

Binary occurrence vectors are constructed, with one axis containing each of the dot-plot symbols used and the other each node in the molecule (numerals are used to refer to the nodes of molecule I, and lower case letters for II). A bit is set only in those vectors whose symbol corresponds to the label of the node as shown in Figure 1. A convenient aspect of this notation is that the symbols may be set equivalent for the purposes of comparison by performing the inclusive OR of the two vectors; allowing "L" (methylene) and "Y" (methine), and also "M" and "N" (secondary and tertiary nitrogen), to be equivalent yields the simplified occurrence vectors shown in Figure 2. This provides a convenient technique for introducing a wide variety of generality into the structure designation.

A compatibility table is next constructed from the binary occurrence vectors of the two molecules whose structures are to be compared. In this table each pair of equivalent nodes of the two molecules will be compared with all other of their equivalent node pairs to ascertain if the nodes of molecule I are structurally related in the same way that the nodes in molecule II are related. This table is a binary symmetrical matrix of area $d \times d$, where $d = \Sigma_i m_i n_i$, and m_i and n_i represent the number of nodes of type i in compounds I and II, respectively, with summation over all different types of nodes common to I and II. Figure 3 shows the compatibility table constructed for the simplified example in Figure 2, for which $d = 1 \cdot 1 + 1 \cdot 1 + 3 \cdot 3 = 11$. To construct the compatibility table a $d \times d$ area in memory is laid out with all positions set, and a cleared position introduced for each of the *incompatibilities* which are derived. The coordinate labels of the table are ordered first by their node from molecule I and then by their node from molecule II; thus for Figure 3 these are $(1,a)$, $(2,b)$, $(3,c)$, $(3,d)$, $(3,e)$, and so forth.

Positions in the compatibility table are now cleared which are found to represent incompatibilities. Table positions which correspond to the mapping of a node in either molecule upon itself are set to zero; for example, in Figure 3 the position $(3,c)$:$(3,d)$ is cleared as it is impossible for nodes c and d in II to be related in the same way that node 3 relates to itself in I. Similarly, position $(4,e)$:$(5,e)$ is cleared because this maps node e of II onto itself. The theoretical justification of clearing such "k-covers" is rigorously treated by Levi.[7] (Note that the table is symmetrical, so that only one set of off-diagonal elements need be saved and examined; for convenience, the full table will be discussed here.)

Connectivity Information. Next the compatibility table information is restricted further according to the molecular connectivities. A



	N	L	A	A	A	M	Y	A	A	A
	1	2	3	4	5	a	b	c	d	e
A:	0	0	1	1	1	0	0	1	1	1
L:	0	1	0	0	0	0	0	0	0	0
M:	0	0	0	0	0	1	0	0	0	0
N:	1	0	0	0	0	0	0	0	0	0
Y:	0	0	0	0	0	0	1	0	0	0

Figure 1. Node numbering, dot-plot symbols, and binary occurrence vectors for dimethylethylamine and methylisopropylamine.

	1	2	3	4	5	a	b	c	d	e
A:	0	0	1	1	1	0	0	1	1	1
L, Y:	0	1	0	0	0	0	1	0	0	0
M, N:	1	0	0	0	0	1	0	0	0	0

Figure 2. Binary occurrence vectors for the compounds of Figure 1 in which L and Y, and also M and N, are made equivalent.

	(1,a)	(2,b)	(3,c)	(3,d)	(3,e)	(4,c)	(4,d)	(4,e)	(5,c)	(5,d)	(5,e)
(1,a)	0	1	1	1	1	1	1	1	1	1	1
(2,b)	1	0	1	1	1	1	1	1	1	1	1
(3,c)	1	1	0	0	0	0	1	1	0	1	1
(3,d)	1	1	0	0	0	1	0	1	1	0	1
(3,e)	1	1	0	0	0	1	1	0	1	1	0
(4.c)	1	1	0	1	1	0	0	0	0	1	1
(4,d)	1	1	1	0	1	0	0	0	1	0	1
(4.e)	1	1	1	1	0	0	0	0	1	1	0
(5,c)	1	1	0	1	1	0	1	1	0	0	0
(5,d)	1	1	1	0	1	1	0	1	0	0	0
(5.e)	1	1	1	1	0	1	1	0	0	0	0

Figure 3. The compatibility table derived from the data of Figure 2 in which the "*k*-cover" positions are cleared.

"neighbor list" and a "degree list" are generated; the former tabulates the node locations (numeral or lower case letter) of the nearest neighbors of all nodes in each molecule, and the degree list shows the number of such neighbors for each node. In molecule I (Figure 1), node 1 is of degree 3 and has neighbors 2, 4, and 5. The "pathfinder", an extension of a typical shortest path algorithm,[16] is used to generate the complete set of maximal paths from any node in the molecule; a path is any route for traversing the nodes in the molecule without going through any node more than once. This list should then contain the largest possible substructures of the two molecules, as illustrated in Figure 4 for compounds I and II. This path information is then used to check each position which is still set in the compatibility table (Figure 3). Each set position in the table is described by coordinates corresponding to pairs of nodes in molecules I and II. If paths identical in their connecting nodes can be found between each of these pairs, the position in the table remains set; otherwise it is cleared. For example, for the position (3,*c*):(5,*d*) the path 3-2-1-5 is equivalent to *c*-*b*-*a*-*d*, as these represent A-L-N-A and A-Y-M-A, respectively; this position remains set in the resulting optimized compatability table, Figure 5. For the position (3,*c*):(5,*e*) there are no equivalent paths in Figure 4, as 3-2-1-5 is not identical with *c*-*b*-*e*, and so the corresponding position is cleared in the Figure 3 table. In the execution of this algorithm, valid positions are verified more rapidly than invalid ones are cleared, and execution is made more efficient by ordering the paths by symbols and by size.

Maximal Common Substructures. The resulting optimized compatibility table (Figure 5) is now analyzed for the longest node-string paths (the optimal substructures) using the algorithm outlined in Figure 6. Because no details on the compatibility table analysis were given in the original description,[7] its operation will be illustrated with the data of Figure 5.

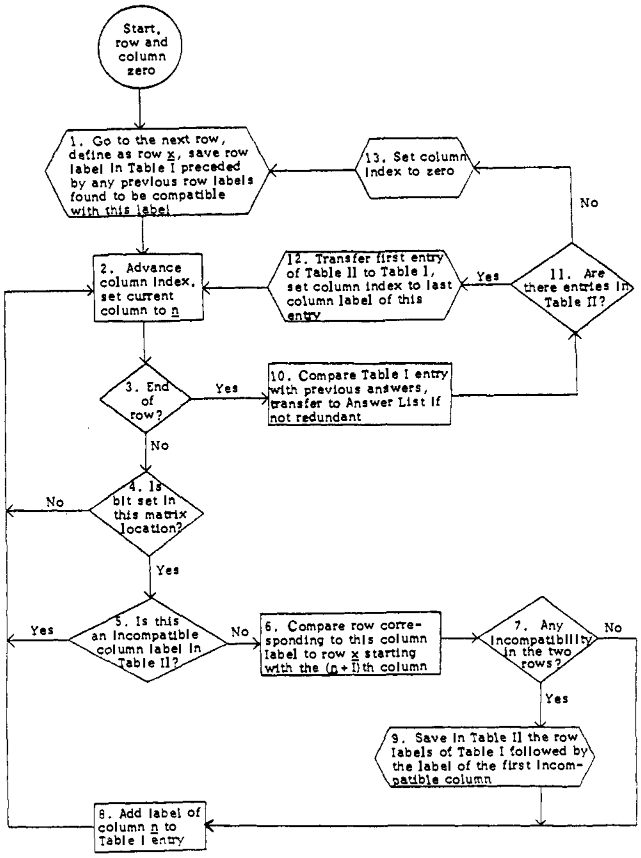
Paths for dimethylethylamine: 1-4, 1-5, 1-2-3, 2-3, 2-1-4, 2-1-5, 3-2-1-4, 3-2-1-5, 4-1-5, 4-1-2-3, 5-1-4, 5-1-2-3

Paths for methylisopropylamine: a-d, a-b-c, a-b-e, b-c, b-e, b-a-d, c-b-e, c-b-a-d, d-a-b-c, d-a-b-e, e-b-c, e-b-a-d

Figure 4. Maximum paths generated for the compounds of Figure 1.

	(1,a)	(2,b)	(3,c)	(3,d)	(3,e)	(4,c)	(4,d)	(4,e)	(5,c)	(5.d)	(5,e)
(1,a)	0	1	1	0	1	0	1	0	0	1	0
(2,b)	1	0	1	0	1	0	1	0	0	1	0
(3,c)	1	1	0	0	0	0	1	0	0	1	0
(3,d)	0	0	0	0	0	0	0	0	0	0	0
(3,e)	1	1	0	0	0	0	1	0	0	1	0
(4,c)	0	0	0	0	0	0	0	0	0	0	0
(4.d)	1	1	1	0	1	0	0	0	0	0	0
(4,e)	0	0	0	0	0	0	0	0	0	0	0
(5,c)	0	0	0	0	0	0	0	0	0	0	0
(5,d)	1	1	1	0	1	0	0	0	0	0	0
(5,e)	0	0	0	0	0	0	0	0	0	0	0

Figure 5. Optimized compatibility table for the data of Figure 2 with positions cleared according to both the "*k*-cover" and maximum path information.



Start, row and column zero
1. Go to the next row, define as row x, save row label in Table I preceded by any previous row labels found to be compatible with this label
2. Advance column index, set current column to n
3. End of row?
Yes
No
4. Is bit set in this matrix location?
No
Yes
5. Is this an incompatible column label in Table II?
Yes
No
6. Compare row corresponding to this column label to row x starting with the (n + 1)th column
7. Any incompatibility in the two rows?
No
Yes
8. Add label of column n to Table I entry
9. Save in Table II the row labels of Table I followed by the label of the first incompatible column
10. Compare Table I entry with previous answers, transfer to Answer List if not redundant
11. Are there entries in Table II?
Yes
No
12. Transfer first entry of Table II to Table I, set column index to last column label of this entry
13. Set column index to zero

Figure 6. Flow chart of algorithm for identifying the longest node strings from the optimized compatibility table.

After start, instruction 1 defines row (1,*a*) as row *x* and places (1,*a*) in Table I; 2 defines column (2,*b*) as n. Instructions 3, 4, and 5 give "no", "yes", and "no", respectively; there are no entries as yet in Table II. Instruction 6 causes row (1,*a*) to be compared to row (2,*b*) starting with column (3,*c*); they are identical, so 7 is "no", and 8 increases the Table I entry to (1,*a*)-(2,*b*). Instruction 2 now sets (3,*c*) as column n, and a similar advance to 6 compares row (1,*a*) with row (3,*c*) starting at column (3,*d*), finding (3,*e*) as the first incompatible column. Thus by instruction 9 the string (1,*a*)-(2,*b*)-(3,*e*) is stored in Table II, and by 8 the Table I string is increased to (1,*a*)-(2,*b*)-(3,*c*).

(1,a)-(2,b)-(3,c)-(4,d)
(1,a)-(2,b)-(3,e)-(4,d)
(1,a)-(2,b)-(3,c)-(5,d)
(1,a)-(2,b)-(3,e)-(5,d)

Figure 7. Answer list: substructures (node-string paths) which were found to be common to dimethylethylamine and methylisopropylamine.

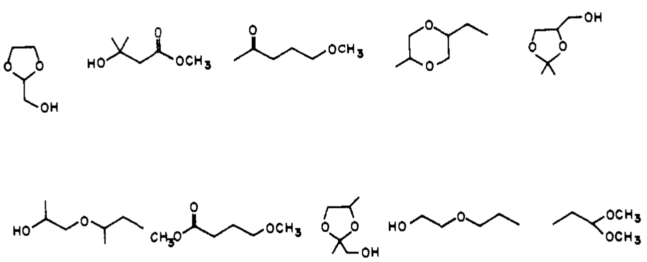


Figure 8. Best-matching reference compounds found by STIRS (MF11.0) for an unknown ketal.

Instruction 2 now sets column (3,*d*) as n, but 4 is "no", so column (3,*e*) is set as n. However, instruction 5 finds that (3,*e*) is an incompatible column label in Table II, so 2 advances n to (4,*c*), and again by instruction 4 advances n to (4,*d*). In 6 the comparison of row (1,*a*) with row (4,*d*) starting at column (4,*e*) finds an incompatibility at (5,*d*), so 9 adds to Table II the string (1,*a*)-(2,*b*)-(3,*c*)-(5,*d*) as its second entry, and 8 increases the Table I entry to (1,*a*)-(2,*b*)-(3,*c*)-(4,*d*). Instructions 2 and 4 now advance the column index through (4,*e*) and (5,*c*) to (5,*d*); instruction 5 finds this as an incompatible column label in Table II. The column index is advanced to (5,*e*), whose bit is not set, so the next column index advance is found to be at the end of the row by instruction 3. Instruction 10 now transfers (1,*a*)-(2,*b*)-(3,*c*)-(4,*d*) from Table I to the answer list (Figure 7), and 12 transfers (1,*a*)-(2,*b*)-(3,*e*) to Table I and sets the column index to (3,*e*). Instruction 1 makes n = (4,*c*), whose bit is not set, so 4 and 1 make n = (4,*d*); 6 again compares (1,*a*) with (4,*d*); now the (5,*d*) incompatibility causes 9 to add (1,*a*)-(2,*b*)-(3,*e*)-(5,*d*) as a second entry to Table II, and 8 to increase the Table I entry to (1,*a*)-(2,*b*)-(3,*e*)-(4,*d*). Instructions 2 and 4 again advance the column index to (5,*d*) which 5 finds as an incompatible column label in Table II. The end of the row is again reached, 10 transfers (1,*a*)-(2,*b*)-(3,*e*)-(4,*d*) to the answer list (Figure 7), 12 transfers (1,*a*)-(2,*b*)-(3,*c*)-(5,*d*) to Table I and sets n = (5,*d*). Instruction 5 finds (5,*d*) as an incompatible column label in Table II, so the instructions cycle again to 10, which places (1,*a*)-(2,*b*)-(3,*c*)-(5,*d*) in the answer list, followed by 12 moving (1,*a*)-(2,*b*)-(3,*e*)-(5,*d*) from Table II to Table I. The column index of (5,*d*) set by 12 is advanced to (5,*e*) by 1, which is cycled again to "end of row", with 10 moving (1,*a*)-(2,*b*)-(3,*e*)-(5,*d*) from Table I to become the fourth answer list entry. Now instruction 11 finds no entries of Table II, and so proceeds to instruction 13 which sets the column index to that of the diagonal element of the row, (2,*b*), defined here as the zero position; the matrix is symmetrical, so that only the upper half of the off-diagonal elements need be used. Instruction 1 sets x = (2,*b*) and enters (1,*a*)-(2,*b*) in Table I, as (1,*a*) was found to be compatible with (2,*b*) when row (1,*a*) was examined by the program. Row (2,*b*) is now compared to row (3,*c*) starting at column (3,*d*); the first incompatibility is at (3,*e*), so (1,*a*)-(2,*b*)-(3,*e*) is entered in Table II and (1,*a*)-(2,*b*)-(3,*c*) in Table I. Columns (3,*d*) and (4,*c*) do not have bits set in row (2,*b*), and (3,*e*) is an incompatibility in Table II, so the next comparison is row (2,*b*) with row (4,*d*). Its incompatibility at (5,*d*) produces (1,*a*)-(2,*b*)-(3,*c*)-(5,*d*) as the second entry in Table II and increases the Table I string to (1,*a*)-(2,*b*)-(3,*c*)-(4,*d*). Instructions 2 and 4 advance the column index to (5,*d*), which is a Table II incompatibility, so the program loops through instructions 10 and 12. The Table I string (1,*a*)-(2,*b*)-(3,*c*)-(4,*d*) is not transferred to the answer list, however, as it is redundant. Continuation of the program follows the same procedure to finish this and subsequent rows; all new strings found are also redundant, so that the answer list remains the same (Figure 7) as after row (1,*a*) was processed.

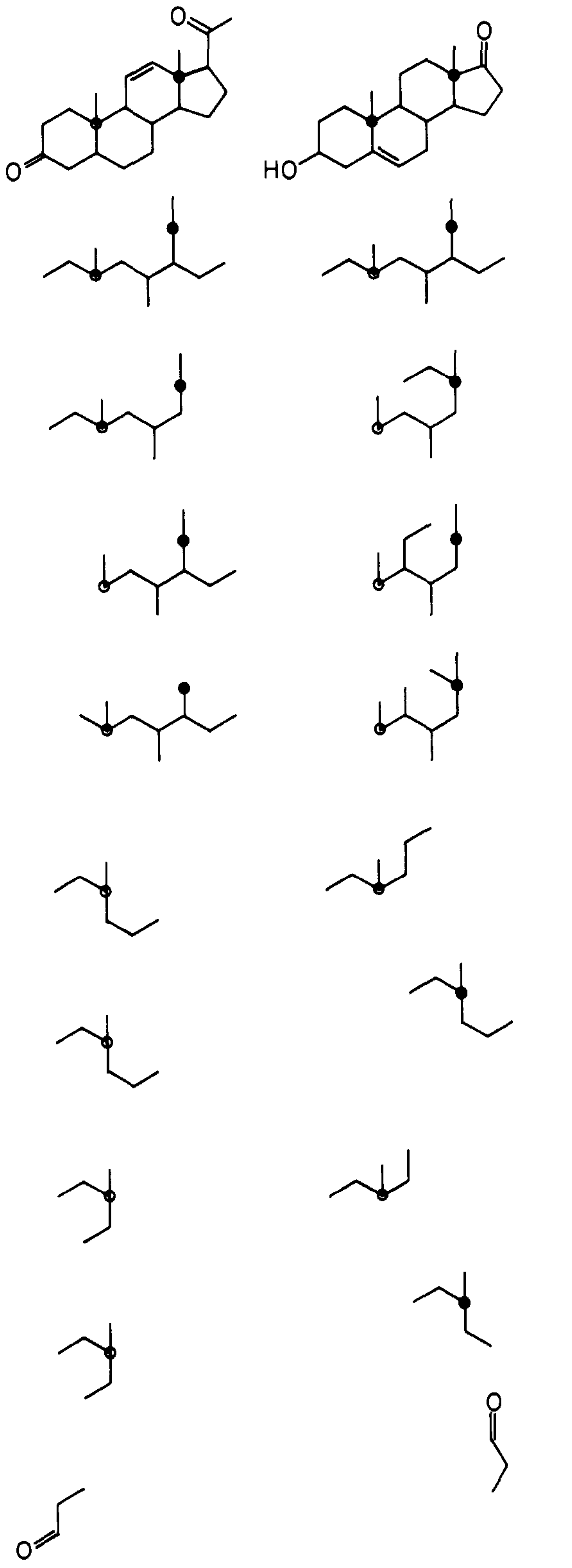


Figure 9. Identical substructure pairs containing ≥six carbons or an oxygen found for the steroids shown. For clarity the fully substituted C-10 and C-13 atoms are indicated by open and closed circles, respectively.

Because different nodes can have the same dot-plot symbols, different string descriptions in the answer list can represent the same chemical substructure. For the STIRS application it was desirable to eliminate such redundancies, which can be done by converting the node designations to dot-plot symbols, and comparing these strings using the allowed equivalencies. For the example, all four strings of the answer list correspond to (M,N)-(L,Y)-A-A.

Results

The structures of a variety of compounds have been compared by the program to generate all possible substructures containing more than two dot-plot symbols. Careful manual comparison of the same compound pairs found no additional substructures. As an example, the structures of Figure 8 represent the ten best-matching compounds found (MF11.0) by STIRS for the mass spectrum of an unknown ketal. In comparing the 45 possible structure pairs, the program found 70 substructures common to both pair members; the largest (7 symbols) substructure found was $CH_3CHCH_2OCHCH_2CH_3$ common to 2-hydroxypropyl *sec*-butyl ether and 2,2-dimethyl-4-hydroxymethyl-1,3-dioxolane. Allowing the L and Y symbols to be equivalent gave 177 such substructures. Generation of these required 0.2-s calculation time per pair using a DEC PDP-11/45 computer.

Calculation requirements increase rapidly with structural complexity, especially with polycyclic fused-ring structures. Comparison (allowing no dot-plot equivalencies) of 28 pairs of steroids required an average of 100 s. For the example of Figure 9 the program found 19 substructures to be common to the pair; those substructures containing more than five carbons, or an oxygen, are shown. When the same problem was presented to chemists, most had difficulty in identifying all of those substructures. Thus the program in its present form appears to have substantial utility for detailed comparison of molecular structures in such problems as spectral interpretation (e.g., STIRS), structure/activity correlations, and synthesis design.

Possible Improvements. Reduction in the time requirements of the program would be of obvious benefit. For molecules such as fused-ring systems which can display complex symmetries, >99% of the computational time requirements are due to the multiplicity of possible paths introduced by each tertiary and quaternary carbon atom. A high proportion of these paths are redundant, and they necessitate extensive comparisons. These problems are similar to those encountered by Lederberg and co-workers[4,17] in development of the DENDRAL cyclic structure generator. Proper heuristics should minimize the path degeneracies; the vertex graph and pruning methods of DENDRAL should substantially reduce the number of nodes to be considered.

For most structural comparison applications it is advantageous that the system be flexible in its ability to introduce structural equivalencies. In the present system this can be done conveniently for dot-plot notations, such as setting "L" (methylene) and "Y" (methine) equivalent. A "ring perception" algorithm[4,17] which would allow a particular complex substructure such as a steroid skeleton to be recognized and treated as a single node would not only shorten calculation times but could also expedite comparisons. For example, the mass spectra of meta- and para-substituted aromatic compounds usually are closely similar; thus for the STIRS application it would be advantageous if the algorithm could treat such structures as equivalent. Work on such improvements is in progress.[18,19]

Acknowledgments. We are grateful to the Environmental Protection Agency for generous financial support, and to the National Science Foundation for partial funding of the PDP-11/45 computer used in this study.

References and Notes

(1) Computer-Aided Interpretation of Mass Spectra. 20. For part 19, see I. K. Mun, R. Venkataraghavan, and F. W. McLafferty, *Anal. Chem.*, in press.

(2) L. C. Ray and R. A. Kirsch, *Science*, **126**, 814 (1957).

(3) E. H. Sussenguth, Jr., *J. Chem. Doc.*, **5**, 36 (1965).

(4) J. Lederberg, *Am. Math. Mon.*, **74,** 522 (1965); J. Lederberg, Dendral-65, Part II, "Topology of Cyclic Graphs", NASA CR-68898, Star No. N66-14074; J. Lederberg, G. L. Sutherland, B. G. Buchanan, E. A. Felgenbaum, A. V. Robertson, A. M. Duffield, and C. Djerassi, *J. Am. Chem. Soc.*, **91**, 2973 (1969); B. G. Buchanan, D. H. Smith, W. C. White, R. J. Gritter, E. A. Feigenbaum, J. Lederberg, and C. Djerassi, *ibid.*, **98**, 6168 (1976).

(5) D. G. Corneil and C. C. Gotlelb, *J. Assoc. Comput. Mach.*, **17**, 51 (1970).

(6) 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, p 72.

(7) G. Levi, *Calcolo*, **9**, 341 (1972).

(8) E. Meyer in "Computer Representation and Manipulation of Chemical Information", W. T. Wipke et al., Ed., Wiley, New York, N.Y., 1974, p 105.

(9) J. E. Armitage and M. F. Lynch, *J. Chem. Soc. C,* 521 (1967); J. E. Armitage, J. E. Crowe, P. N. Evans, M. F. Lynch, and J. A. McGuirk, *J. Chem. Doc.*, **7,** 209 (1967).

(10) J. M. Harrison and M. F. Lynch, *J. Chem. Soc. C*, 2082, (1970).

(11) K.-S. Kwok, R. Venkataraghavan, and F. W. McLafferty, *J. Am. Chem. Soc.*, **95**, 4185 (1973); H. E. Dayringer, G. M. Pesyna, R. Venkataraghavan, and F. W. McLafferty, *Org. Mass Spectrom.*, **11**, 529 (1976).

(12) E. J. Corey, W. T. Wipke, R. D. Cramer, III, and W. J. Howe, *J. Am. Chem. Soc.*, **94**, 421, 431 (1972).

(13) A. J. Stuper and P. C. Jurs, *J. Am. Chem. Soc.*, **97**, 182 (1975).

(14) E. Hyde, F. W. Matthews, L. H. Thomson, and W. J. Wiswesser, *J. Chem. Doc.*, **7**, 200 (1967).

(15) H. E. Dayringer, Ph.D. Thesis, Cornell University, 1976.

(16) N. Deo, "Graph Theory with Applications to Englneering and Computer Science", Prentice-Hall, Englewood Cliffs, N.J., 1974, p 297.

(17) D. H. Smith, L. M. Masinter, and N. S. Sridharan in "Computer Representation and Manipulation of Chemical Information", W. T. Wipke et al., Ed., Wiley, New York, N.Y., 1974, p 287.

(18) Listings of the described programs are available in PDP-11/45 assembler language from the authors.

(19) Dr. R. G. Dromey has polnted out additional pertinent work: H. G. Barrow and R. M. Burstall, *Inf. Process. Lett.,* **4**, 83 (1976).

Structure Elucidation with Lanthanide-Induced Shifts. 2. Conformational Analysis of Cyclohexanecarbonitrile

Douglas J. Raber,* Milton D. Johnston, Jr.,* and Michael A. Schwalke

Contribution from the Department of Chemistry, University of South Florida, Tampa, Florida 33620. *Received April 26, 1977*

Abstract: The molecular structure of cyclohexanecarbonitrile, a conformationally mobile molecule, has been examined in solution with the aid of a lanthanide shift reagent. Nonlinear regression analysis of the NMR data obtained in the presence of $Eu(fod)_3$ afforded the bound shifts of the LS complex. A priori calculation of the bound shifts for each of the two chair conformations using a parametrized form of the pseudocontact equation results in the conclusion that cyclohexanecarbonitrile exists in the conformation with an equatorial cyano group to the extent of 54%, in good agreement with previous work. Agreement between observed and calculated shifts for this distribution of conformers is excellent.

The rigorous determination of molecular structure is a problem of great significance in chemistry and is of particular importance for the case of conformationally flexible molecules in the liquid state. However, such structural information frequently has not been readily available. While NMR spectroscopy has proven to be the most powerful tool for confor-