Molecular Structure Disassembly Program (MOSDAP): A Chemical Information Model To Automate Structure-Based Physical Property Estimation

John W. Raymond*

Department of Biophysics, University of Michigan, 930 N. University,
Ann Arbor, Michigan 48109-1055

Tony N. Rogers[†]

Department of Chemical Engineering, Michigan Technological University (MTU),
Houghton, Michigan 49931-1295

Received July 29, 1998

Chemical information theory and molecular structure searching have long been used as computational aids to researchers in the pharmaceutical field to estimate molecular structure−property relationships and to assist in drug design. Tailored to these and other specific applications, such endeavors have been expensive to develop and typically are very specialized. Often, they are not readily available and are not a part of the open literature. Because the number of chemicals in commercial use is growing daily (with over 18 million molecular species now catalogued by Chemical Abstract Services), there is a need among engineers in the chemical process industries for predictive structure−property algorithms. The most common and useful methods are those based on group contribution that require only the chemical structure of interest. Unfortunately, each group contribution method typically has its own fragment library and specialized rules, making such models difficult to automate for general use by the engineering community. This work, which has culminated in the creation of the Molecular Structure Disassembly Program (MOSDAP) software, is focused on combining and improving upon the best published methods in four areas: (1) lexicographical entry of structures, (2) prescreening methods, (3) abstract representation of molecular structures, and (4) structure manipulation routines. Additional features, such as a custom modification of the published Ullman substructure search algorithm specific to molecular graphs and an exact cover procedure to elucidate structural ambiguities, have been added by us to address specific problems encountered in group contribution methods. At present, most of the popular published group contribution methods can be automated using MOSDAP as a general engine for converting formula line notation (e.g., SMILES strings) into corresponding sets of functional groups and/or features.

MOTIVATION FOR DEVELOPING MOSDAP

Over 18 million molecular compounds are currently catalogued in the CAS[1] registry, and the number of chemicals in use by industry or on regulated lists is increasing each year. Consequently, physical properties for these chemicals are urgently needed by engineers for calculations involving process design, mass and energy balances, pollution control, and fate and transport in the environment. Because of the expense and the number of chemicals involved, direct experimental measurement of properties is infeasible. Predictive group contribution or structure−property methods are very helpful, but they are frequently quite different in their specific calculations and rules.

This research effort has as its goal the automation of group contribution (GC) and structure−property methods for predicting chemical properties. By borrowing and tailoring concepts from chemical information science, graph theory, and pharmaceutical research, as well as adding contributions of our own, we have created software titled Molecular Structure Disassembly Program (MOSDAP). It has the potential to become a central algorithm that ties together the common elements of many GC methods and quantitative structure property relationships (QSPRs) now employed by engineers and scientists. Starting with a two-dimensional representation (graph projection or line notation) of the molecule of interest, the MOSDAP program first converts the representation into an abstract adjacency listing. This representation may then be manipulated to look for the presence of certain moieties, structural features, or collections of atoms and bonds. An output listing of such groups is then a direct input to a selected structure−property estimation algorithm.

As an example of automating the Benson[2] GC method, a popular commercial drawing program such as ChemDraw[3] can be used to generate a graphical structure of a molecule and convert it to SMILES notation. Then, MOSDAP will convert the SMILES string into the internal representation for a chemical structure, search it for the presence of Benson groups, and output a list of Benson group occurrences in the original chemical.

Through substructure searching techniques, it is possible to (1) identify specific functional groups and features (e.g.,

* To whom correspondence should be addressed. E-mail: jwraymon@umich.edu.

[†] E-mail: tnrogers@mtu.edu.

10.1021/ci9803334 CCC: $18.00 © 1999 American Chemical Society
Published on Web 04/10/1999

estimate the biodegradability of a chemical based on the presence or absence of certain atoms or functional groups[4]) or (2) exhaustively account for all valid combinations of groups of which a molecule is comprised. MOSDAP is designed to support both approaches so as to be applicable to the vast majority of existing GC property models.

PROPERTY ESTIMATION FROM CHEMICAL STRUCTURES

The estimation of physical and thermodynamic chemical properties, of critical importance in chemical process design, has been attempted using a wide variety of techniques, but among the most prevalent and widely applicable of the methods is the group contribution[5−9] technique. This type of correlation allows for the estimation of chemical properties with only an understanding of the chemical structure. GC methods employ a presumed cumulative macroscopic effect due to the presence of substructure fragments and structural features. GC methods can be categorized into three general classifications: (1) methods based simply on the presence of specified substructure fragments or molecular features (e.g., Boethling); (2) structural increment methods which rely on the complete reduction of the molecular structure into a pool of specified substructures (e.g., UNIFAC[10]); and (3) hybrid methods which rely on the structural increment technique for initial property estimation but also employ correction factors based upon the presence of pertinent molecular features and additional structural moieties (e.g., Pintar[11]).

Although significant effort has been expended in developing structure−property relationships, the applicable subfragments are often based only on the creator's intuition of which subfragments are relevant, resulting in methods that are occasionally ambiguous and potentially have multiple possible subfragment groupings. These disadvantages are further exacerbated by subsequent modifications to the original set of subfragments in an effort to extend the applicability of the estimation method. With the aid of an automated fragmentation program, these limitations can be addressed while allowing for a reliable statistical analysis during GC development.

AUTOMATION OF GROUP CONTRIBUTION METHODS

To alleviate much of the effort involved in using these increasingly complex estimation methods and to allow for the more systematic development of future correlations, an attempt has been made to design a robust, computerized platform for the automation of all GC determinations, the Molecular Structure Disassembly Program (MOSDAP). This goal of a universal group management program, however, is not new. Early published efforts include the works of Brasie[12] and Jochelson.[13] The Brasie attempt was based upon a specialized lexicographical analysis of the Wiswesser line notation (WLN)[14] for chemical structure representation. This effort, although pioneering, is extremely limited because its searching mechanism was performed directly on the WLN chemical structure string rather than on a representation of the molecular graph. This process is computationally inefficient and severely limits the applicability of the substructure fragments that can be represented. Although the Jochelson approach was more theoretically grounded, it also suffered limitations primarily due to inaccuracies in its substructure searching mechanism and its unwieldy input structure. Although the accuracy limitations inherent in the original substructure search algorithm[15] were later addressed,[16,17] the Jochelson attempt does not appear to have been revisited.

More recent published efforts include the work of Adams,[18] Qu,[19] and Drefahl.[20] The Adams paper primarily presented an approach to determine the most reliable and accurate combination of various GC methods for a given chemical and set of required properties, facilitated by an automated approach. This effort employed an inconvenient "drawing" method for structural input whereby the query structures were drawn using alpha-numeric keyboard symbols at the command line. Substructure searching was accomplished using a modification of the set reduction algorithm due to Figueras,[21] and the ring perception mechanism was unspecified. Although this effort appears to be the most advanced direct attempt at GC automation, the adopted search syntax is inconvenient and lacks versatility, and the substructure search algorithm has the potential to make false detections.[16]

Qu's effort employs the Advanced Encoding System (AES),[22] a modified WLN, to represent chemical structures and relies on a simple graph walking routine to search for chemical substructures. This work, like Brasie's effort, is handicapped by the cryptic and restrictive structure notation and also from an unsophisticated search algorithm. DESOC, the Drefahl program, uses a relatively straightforward modified Simplified Molecular Input Line Entry System (SMILES)[23−25] as the chemical entry syntax, but it is was developed primarily to determine the similarities and dissimilarities between chemical structures for extrapolating physical properties based on the maximum common subgraph (MCS)[26,27] concept rather than performing a direct analysis using the GC technique.

Commercial GC-related software has also been made available for use in the automation of group contribution correlations,[28−30] but these products frequently support only a limited number of GC methods and typically do not allow the user to modify or create new substructure libraries if new or improved correlations are published. Other commercially available programs require the user to manually construct the query molecule via a menu of valid substructures for each property estimation method.[31−33] None of these products were designed specifically to aid the researcher in determining an appropriate set of subfragments. Unlike the other commercially available programs, the Cranium program[30] does permit user-defined GC methods by means of a graphical substructural fragment declaration. Cranium is user-friendly and represents a useful tool for GC use, but it appears to be primarily oriented toward the presentation of a quality analysis of the data and estimation methods for a given property. Although the graphical interface allows elementary subfragment declaration, it does not offer a sophisticated enough representation of substructures to be used with many published GC methods (e.g., Sugden, Klopman, etc.[9]) or, more importantly, for future GC development employing increasingly complex substructure declarations. The algorithm employed in the substructure search appears to be a simple graph walking routine.[30] As with the published attempts, none of the commercial programs allow

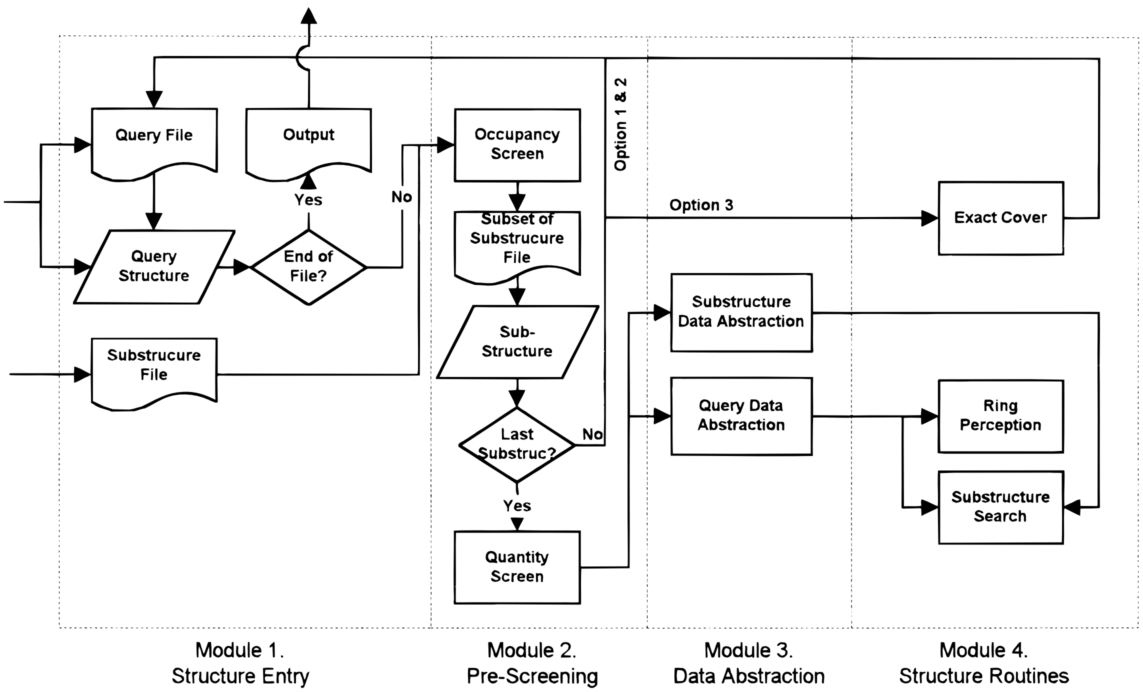


Figure 1. MOSDAP process scheme. Search options: (1) nontruncating, sequential; (2) truncating, sequential; (3) truncating, combinatorial.

for logically constrained subfragments or the detection of nonsubfragment structural features specified by the user.

MOSDAP MODULAR OPERATIONS

The MOSDAP system is compartmentalized into four distinct operations or "modules". These include the lexicographical input of the query and subfragment structures (module 1), prescreening the structures to preempt unnecessary parsing and searching (module 2), parsing the structures into the appropriate internal data types (module 3), and the structural manipulation routines (module 4). These operations were compartmentalized so that any one of them can be augmented while minimizing the necessity to revise existing code. Figure 1 presents a simplified schematic of the relationship between the specific modules employed by MOSDAP. Once the structural analysis is performed on the query structure(s) to determine which subfragments or structural features are present and in what combinations and quantities, they are output as vectors containing the structural feature identification and the quantity of detections.

Module 1: Structure Entry System. The chemical structure entry system and lexicographical conversion process employed by MOSDAP has been designed in a completely modular format reminiscent of other published lexicographical conversion procedures.[34−36] Therefore, all structure manipulation procedures are implemented independently of the chemical structure entry system. This format was developed so that future implementations of MOSDAP can accommodate chemical structure entry from other linear chemical syntaxes or directly from a graphical interface. The chemical structure entry system used by this program consists of a single query structure or query structure file and a substructure file. Since the substructure fragment file is external to the program, it can be modified without any knowledge of the internal operation of the MOSDAP program. Additional GC methods can be analyzed by simply creating new subfragment data files and having the MOSDAP program return whether the fragmentation procedure was successful along with the respective quantities of the identified detections.

The chemical structure input syntax was chosen based on three criteria: (1) the syntax must be simple, intuitive, and directly amenable to fragmentation; (2) it must be as concise as possible; and (3) it must be adaptable to computerized manipulation. The field of chemical syntax encoding is surprisingly diverse; therefore, a wide variety of substructure representation methods was available for evaluation, including Wisswesser line notation (WLN),[14] Advanced Encoding System[22] (AES), Dyson IUPAC,[37,38] Simplified Molecular Line Entry System (SMILES),[23−25] SYBYL line notation (SLN),[39] and various others.[40,41] Of these, SMILES appears to be the simplest and most intuitive for use in developing potential substructure search fragments. Although WLN was used by Brasie and AES notation was used by Qu as the entry systems in their respective efforts, they have the disadvantage of being relatively cryptic, making it difficult for users unfamiliar with the syntax to readily modify the substructure fragments.

The proposed substructure fragment syntax used by MOSDAP has been developed as a subset of SMILES nomenclature. It provides a simple mechanism for the enumeration of readily recognizable substructure fragments. To construct the subfragment library, the characters and associated definitions listed in Table 1 are used in conjunction with conventional SMILES nomenclature. The ability to differentiate between atoms that are targeted for extraction upon detection and those present solely for search constraint purposes was the primary focus of the MOSDAP search syntax. An attempt was also made to keep the syntax as simple as possible, as most current GC methods employ a limited quantity of structural identifiers.

Additional features instituted in MOSDAP include routines supporting molecular features and constrained subfragment detection. These features are important in GC methods such

Table 1. MOSDAP Search Declarations

Atom/Bond Symbols	
A	any aliphatic atom
a	any aromatic atom
?	any atom
[X,x]	aliphatic or aromatic atom of type X, where X is represented by an atom's atomic symbol
∼	any bond
search symbols	
*	unknown degree operator; when an atom is preceded by this operator, it is assumed to be of unknown valence structure AND is not extracted from the query structure upon detection of the substructure
!	not operator
≪,≫	compound search operator braces; compd search operators are confined within the double-angle braces; currently only ring constraint operators are supported by MOSDAP; see Table 3 for compd search operator declarations
{,}	search operator grouping braces; these characters are used to group structural moieties within a subfragment dictated by a preceding search operator; unless braces are used with a search operator, the default domain of a search operator is assumed to be only the adjacent atom

as the Franklin[42,43] and Joback[44] methods where subfragments are subject to constraints not directly inherent in the subfragment structure. Currently, MOSDAP only supports bond- and ring-based molecular features and ring-based constrained subfragments. Molecular feature queries are defined by simply confining the compound operator declarations within double-angle braces with no associated subfragment. The MOSDAP interpreter determines whether the expression represents a molecular feature or a constrained subfragment. Table 2 presents a list of molecular feature/ constrained subfragment declarations supported by MOSDAP.

The modified syntax enables a versatile representation of chemical substructures. The syntax can be used to create simple subfragment structures, structurally and logically constrained subfragment structures, and specific structural features. This versatility permits the use of most published GC methods with MOSDAP, unlike other published and commercial GC automation programs which are limited in the types of substructure fragments and molecular features that can be defined. Figure 2 depicts several substructures and their corresponding syntax representations for the type of simple fragments most frequently encountered in typical GC correlations. Figure 3 illustrates the type of substructures and features that MOSDAP can address that the other established programs typically cannot.

Currently, query structures must be entered using valid SMILES notation. This is not foreseen as an impediment because SMILES notation is firmly established in the chemical information literature and large existing databases of SMILES notations are readily available.[45] The syntax is also very intuitive, and graphical commercial software packages[3] allow the generation of SMILES strings directly from a graphical representation of the chemical structure. Additional lexicographical conversion routines can be added as necessary for the translation of query structures independent of the subfragment declaration syntax.

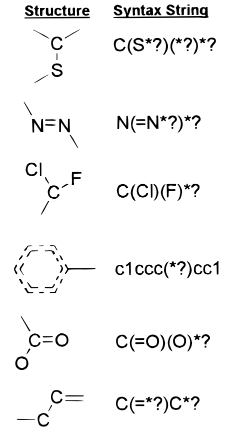
Module 2: Prescreening Procedure. Since the geometric substructure searching for GC methods is a computationally intensive operation, a prescreening procedure is employed to preempt unnecessary substructure searching. Although sophisticated screening strategies have been developed for use with large databases in applications such as patent searches, chemical registry, and pharmacological searches including sophisticated screen set generation methods[46−49] and systems based upon statistical occurrence analysis,[50−53] these advanced screening mechanisms were not directly employed in this instance due to the typically truncating nature of the GC-focused search and the often nominal size of the substructure fragments.

The current prescreening procedure involves essentially two distinct operations: (1) a quantity screen and (2) an occupancy screen. The quantity screening process involves a single-pass scan of each string representing a structure and denoting the quantity of each type of designated structural marker. Markers include the atom and bonding types, the presence of cyclic structures, and certain multiatom complexes. Screening keys were selected based on published statistical occurrence data in typical chemical databases but were further constrained by the routinely small size of the GC method substructures, which precludes the use of most augmented atom complexes as structural screening keys. This information is stored as a bit field denoting the quantity of each key for each query and fragment structure. Prior to proceeding on to the other more computationally demanding procedures, a simple comparison of the quantity of designated keys between a substructure and a query molecule is made to determine whether it is possible for the substructure to be present in the query molecule. As an example, if a substructure possesses two oxygen atoms and the query molecule contains only one, then it is not necessary to proceed with further analysis of this particular subfragment. This process is further constrained during a sequential truncating search by decrementing the appropriate quantity bit fields in the query molecule upon the detection of a substructure. This further constrains the prescreening process, resulting in a more efficient analysis using the sequential extraction procedure.

While the quantity screening process greatly enhances the efficiency of structural analysis and is the sole screening procedure for simple queries, the screening process can be enhanced further for queries of sufficient complexity by introducing order to the substructure file. If MOSDAP determines that the query molecule/file is sufficiently complex and/or the substructure file is large enough, the substructure file is sorted into a series of bins such that the substructures contained in the first bin contain a structural key (k1) and subsequent bins do not contain k1. Substructures in the second bin will contain structural key k2 but not structural key k1, but subsequent bins will not contain k1 or k2. Using this file structure, the screening procedure will first determine if k1 is present in the binary bit field of the query molecule. If it is found to be present, the bin specifying the substructures which possess k1 will be screened by comparing the entire binary bit field of the query molecule with each of the substructure bit fields in the substructure bin by using the bitwise AND operator. If it is found that the bitwise comparison for a particular substructure is a success, then the screening procedure proceeds to the

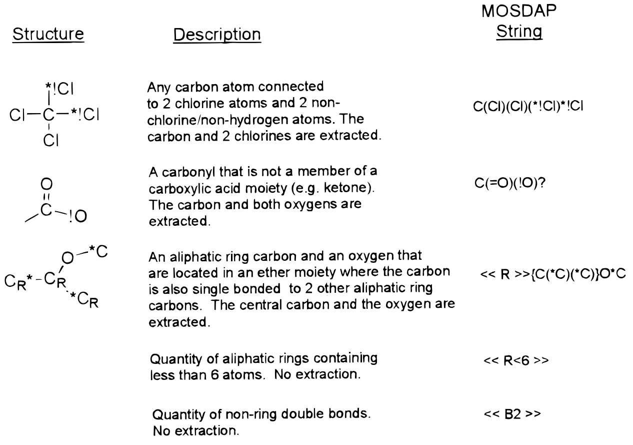
Table 2. Compound Search Operators

bond features	definitions	ring features	definitions
B	any nonring-constrained bond	R	any aliphatic ring-constrained atom
b	any ring-constrained bond	r	any aromatic ring-constrained atom
Bb	unconstrained bond	Rr	unconstrained ring
B#	nonring constrained bond of type denoted by # (i.e., 1, single; 2, double; 3, triple; 4, aromatic; 1, any type)	R#	aliphatic ring with number of members denoted by #
b#	ring-constrained bond of type denoted by #	r#	aromatic ring with number of members denoted by #
Bb#	unconstrained bond of type denoted by #	Rr#	unconstrained ring with number of members denoted by #
		Logical Operators	
>#	greater than operator used with feature type denoted by #	<#	less than operator used with feature type denoted by #
!	not operator used to exclude detections declared with feature type		



Structure	Syntax String
	C(S*?)(*?)*?
	N(=N*?)*?
	C(Cl)(F)*?
	c1ccc(*?)cc1
	C(=O)(O)*?
	C(=*?)C*?

Figure 2. Example substructure representations.



Structure	Description	MOSDAP String
	Any carbon atom connected to 2 chlorine atoms and 2 non-chlorine/non-hydrogen atoms. The carbon and 2 chlorines are extracted.	C(Cl)(Cl)(*!Cl)*!Cl
	A carbonyl that is not a member of a carboxylic acid moiety (e.g. ketone). The carbon and both oxygens are extracted.	C(=O)(!O)?
	An aliphatic ring carbon and an oxygen that are located in an ether moiety where the carbon is also single bonded to 2 other aliphatic ring carbons. The central carbon and the oxygen are extracted.	<< R >>{C(*C)(*C)}O*C
	Quantity of aliphatic rings containing less than 6 atoms. No extraction.	<< R<6 >>
	Quantity of non-ring double bonds. No extraction.	<< B2 >>

Figure 3. Extended example substructure representations.

aforementioned quantity screen. If it was discovered during the query that the molecule did not contain k1, then k2 is investigated, bypassing needless screening operations of the substructures contained within the bin of substructures denoted by k1. This results in a search of a subset of the substructure file rather than the entire file for each query molecule. Figure 1 (module 2) illustrates the procedural progression of the screening process for a complex query.

Module 3: Data Structure Abstraction. Once a structure has passed the prescreening operation, it is parsed into the internal data representation if it has not already been created during a previous iteration. By allocating data structures only as needed, larger queries may be executed more efficiently, requiring fewer iterations and less memory consumption. Currently, only SMILES is supported in the conversion operation. The internal data structure is represented as a modified, dynamic adjacency listing,[54] a frequently used data structure for representing sparse matrixes and graph structures. The adjacency listing consists of a list of vertex nodes, each serving as the head of a separate list of edge nodes. In this implementation, each vertex node represents an atom structure and each edge node represents a bonding between the head vertex node and a target vertex node denoted by an integer location or a pointer to the target location within the vertex array. Each element of the vertex list consists of an object of the type atom class and each edge node consists of an object of the type bond class. The atom and bond classes serve to encapsulate all of the information necessary to represent an atom and a bond, respectively. The adjacency listing technique for representing a chemical structure has been previously presented[55] in the literature. Figure 5 presents a simplistic illustration of this representation for the example query and subfragment structures presented in Figure 4.

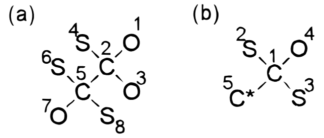


Figure 4. (a) Query graph. (b) Subgraph.

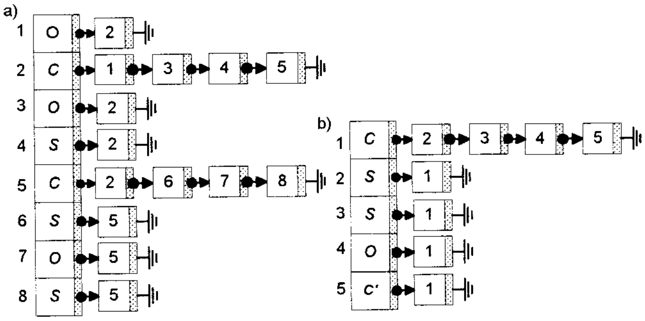


Figure 5. (a) Query adjacency listing, α. (b) Substructure adjacency listing, *β*.

For the purposes of this implementation, the concepts of inheritance and encapsulation offered by C++ were employed to represent the chemical structures and search entities. The chemical system and molecular, atom, and bond classes can be augmented in the future either through direct modification of the elements of the class or through the use of object-oriented class inheritance in a manner suggestive of that proposed by Bauerschmidt.[56] This representation is conducive to the incorporation of three-dimensional properties, stereochemistry representation, and advanced electronic information. It is also directly amenable to the comparative searching involved in the substructure searching process, as each atom object contains and references a considerable amount of information regarding its immediate environment, and it is able to store significantly more information regarding the molecular structure than the standard adjacency matrix while simultaneously preventing the storage of nonentity information. By employing this method of data representation, each node (atom) comparison involved in the substructure search routine actually involves a comparison of an augmented atom complex, hastening the search.[57] Object-oriented programming (OOP)[58] allows modification and improvements to the MOSDAP program without extensive redefinition of the data types or modification to the member functions. It serves to encapsulate the molecular structure information in a data format that is both easy to interpret and adapt.

Module 4: Structure Manipulation Routines. As previously mentioned, different GC methods require varying substructure search techniques. Detecting the presence of a subfragment requires a simple, straightforward subgraph isomorphism procedure without any modification to the query structure, while a structural increment method requires that the query structure be truncated to reflect subfragment detections as they occur. This difference is further exacerbated by the possibility of ambiguous groupings of subfragments that can completely comprise a query structure. While programs such as Cranium employ weighting correlations to order the subfragments in a discriminate order, this procedure is arbitrary and does not alert the user to the presence or extent of subfragment detection ambiguities. It may also result in a failed search even though a valid substructure grouping is present within the subfragment library. This can occur when the extraction of a substructure from the query molecule during a sequential search results in a molecular structure residual that cannot be completely disassembled using the remaining substructures in the subfragment library. The extraction of a different substructure, however, may result in a molecular structure residual that is capable of being disassembled into a subset of the remaining substructures.

MOSDAP avoids this limitation by offering three distinct substructure search options: (1) nontruncating sequential searches; (2) truncating sequential searches; and (3) combinatorial truncating searches. Nontruncating sequential searches involve a scan of the query structures to simply determine which of the subfragment entries are present. This type of search is necessary when the subfragments are conflicting (i.e., when atoms present in one subfragment may also be used in another subfragment) or when a simple determination of whether a subfragment is present is sufficient. Sequential truncating searches are typically used for structural increment GC methods when the subfragments are to be extracted from the molecular structure in the order detected. For this type of search, the subfragment library must not be ambiguous; therefore, atoms from one subfragment are not permitted to be present in another subfragment. This is accomplished via nonredundant subfragments or a strict hierarchical ordering of the substructures such as Cranium's ordering procedure.

The combinatorial truncating search mode is employed when the subfragment inventory is indefinite. This is prevalent in GC methods where there is no established hierarchal order for the detection of substructures and/or portions of the subfragments coincide with other subfragments such as with the UNIFAC method.[10] A dramatic example of the potential for ambiguous groupings is illustrated by 1,2,3-propanetriol triacetate which possesses 15 potential UNIFAC breakdowns using the Hansen[59] revision of the subfragment library (Figure 9). Combinatorial searches are accomplished by first performing a nontruncating sequential search to determine which subfragments are present within the query structure and then applying a set cover procedure[60] on the candidate substructures to determine the combinations of the tentative detections that result in valid

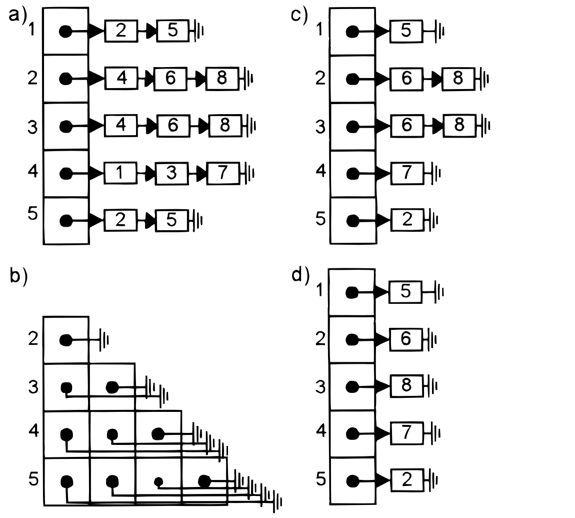


Figure 6. (a) Initial comparison. (b) Initial refinement. (c) Storage structure. (d) Complete search.

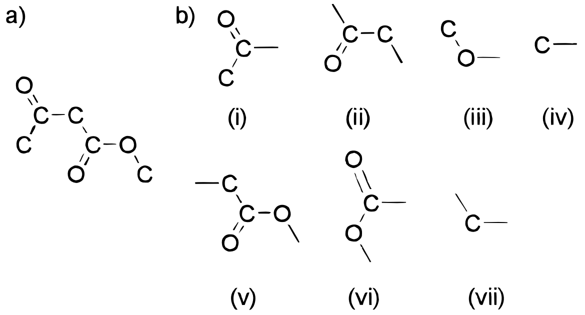


Figure 7. (a) Query molecule. (b) Multicover subgraphs.

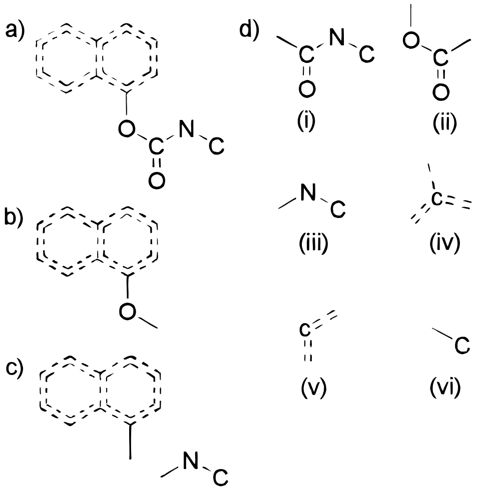


Figure 8. Example sequential search failure.

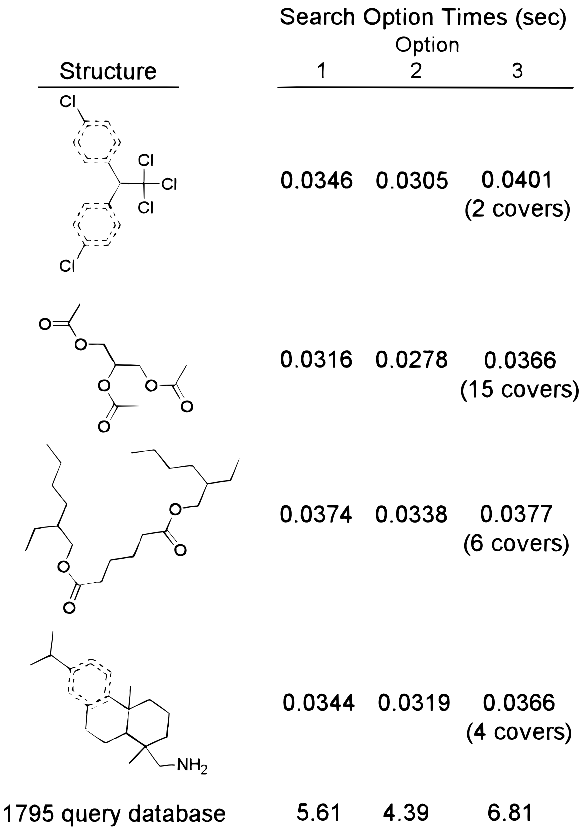

Search Option Times (sec)
Option
Structure
1 2 3
Cl
Cl
Cl
Cl
Cl
0.0346 0.0305 0.0401 (2 covers)
O
O
O
O
O
O
0.0316 0.0278 0.0366 (15 covers)
O
O
O
O
0.0374 0.0338 0.0377 (6 covers)
NH2
0.0344 0.0319 0.0366 (4 covers)
1795 query database 5.61 4.39 6.81

Figure 9. MOSDAP simulation results. Search options: (1) nontruncating, sequential; (2) truncating, sequential; (3) truncating, combinatorial.

groupings. This method provides all possible groupings of subfragments that completely constitute the query structure.

Ring Perception Procedure. Ring perception is employed by MOSDAP for structural feature detection as well as for constrained subfragment searching. The ring feature detection is used in GC methods such as the Lebas[6] molar volume method where information like the quantity of rings containing a specified number of atoms is required. A comprehensive treatise on the subject of ring perception in chemical structures is presented by Downs,[61,62] in which he reviews the theory of ring perception and the body of ring perception algorithms developed up to the publication of his papers.

Ring perception in MOSDAP is accomplished using an algorithm due to Figueras.[55] The Figueras algorithm is a culmination of the algorithms due to Doucet[63] and Balducci.[64] It is based on the node-elimination procedure of Doucet and the breadth-first search (BFS) procedure of Balducci. In comparison trials of the three algorithms, Figueras reveals that his algorithm is considerably more efficient than either the Doucet or the Balducci algorithm. Using the excellent documentation and examples provided by Figueras in his published paper, the ring perception algorithm was added to MOSDAP's molecular structure manipulation routine inventory.

Ring perception is only performed when necessary, as illustrated by Figure 1. In substructures where ring membership is not implicitly represented by the structure (e.g., a benzene ring), it is necessary to label the atoms/bonds in the query molecule at a local level prior to substructure searching. Figure 3 presents a substructure illustrating a ring membership explicitly. It is evident from the substructure that ring membership is an explicit rather than structurally implicit property, and in order to detect this substructure in a query molecule possessing a ring, the members of the ring within the molecule must be identified and labeled prior to initiating the substructure search. MOSDAP interprets from the substructure syntax whether a ring perception operation on the query molecule is necessary and performs the ring perception routine if it has not already been instituted by a previous substructure search operation.

Substructure Search Procedure. Chemical substructure searching, an application of the more general subgraph isomorphism procedure of graph theory, has been addressed considerably in the literature by the chemical information community.[16,65−67] It has been used extensively in areas involved with chemical structure information, such as chemical registry, pharmaceutical lead development, organic synthesis, etc., where the efficient delineation of large numbers of query molecules for pertinent substructural features is of primary importance. These activities, though, have not been directed at the specific requirements necessary for explicitly analyzing chemical structures for GC applications. For the development of MOSDAP, a literature search was performed in an attempt to discern the most efficient algorithm applicable to chemical substructure searching. Several papers have been written espousing various published subgraph isomorphism/chemical substructure searching algorithms,[15,17,21,68−75] but there is a general consensus that the Ullman algorithm[76] is the most appropriate for geometric chemical substructure searching.[16,65−68,77−90] For this application, a dynamic implementation of the Ullman algorithm was employed. The Ullman algorithm is composed of a depth-first search (DFS) and an intermittent relaxation procedure. This process is thoroughly explained in the literature with simple examples provided by Brint[82] and Mitchell.[85]

The dynamic implementation of the search algorithm within MOSDAP is initiated by the construction of the initial match structure. The initial matching at depth zero is generated by comparing each atom object in the subfragment (β) structure to each atom object in the query (α) structure. This information is appended to the match structure dynamically as a series of linked lists indicating the location in the query structure of all potential matches. Since each atom object encapsulates information regarding its immediate vicinity, this is analogous to comparing two approximate augmented atom complexes. For the chemical structures presented in Figure 4, the initial match listings have been appended to the match structure (Figure 6a). If any pointer in the match structure at depth zero remained null during the initial matching, the search would abort. Once a valid initial matching has been constructed, the process reverts to a DFS with intermittent refinement at each depth increment. Note that the actual implementation of the substructure search in MOSDAP is much more efficient than the one presented in this simplistic example.

Refinement. The refinement procedure serves to cull possible matchings from the matching lists at each depth by ensuring that all immediate neighbors of an atom object in the query and substructure graphs also have corresponding potential matches in the match structure and removing any that do not satisfy this requirement, thus abbreviating the DFS. In the originally published Ullman algorithm refinement procedure, this relaxation condition requires only that a potential query node match exists for each subfragment node. This allows for ambiguity if the potential neighbor mappings overlap. This general provision is necessary in large, arbitrary graphs of high degree, but since molecular graphs are bounded and of low degree, the refinement procedure can be further constrained to require an exact 1:1 mapping of neighboring atoms.

This procedure is illustrated with the refinement of the initial matching (Figure 6a). From the matching listings, the first possible matching involves the first atom object in the subfragment structure and the second atom object in the query structure (1,2). For the refinement (1,2), all neighbors of the first subfragment atom, $\beta(1)$, and the second query atom, $\alpha(2)$, are determined (Figure 4). They are $\alpha(1,3,4,5)$ and $\beta(2,3,4,5)$, respectively; therefore, the conceivable candidate matchings in the match structure are (2,1:3:4:5), (3,1:3:4:5), (4,1:3:4:5), and (5,1:3:4:5). From the match listing structure (Figure 6a), only (2,4), (3,4), (4,1), (4,3), and (5,5) are possible matches. It is evident that subfragment atoms $\beta(1)$ and $\alpha(2)$ both specify query atom $\alpha(4)$, and since each β atom can correspond with only one distinct α atom, this particular refinement of element (1,2) has failed. The matching (1,2) is then removed from the match structure. In MOSDAP, ambiguities in such cases are resolved by reverting to a simple DFS of the neighbor candidates to ensure that there is a 1:1 correspondence in circumstances where an ambiguity was detected. This is analogous to a bipartite matching.[91] Although several efficient bipartite matching algorithms have been proposed, a simple DFS approach was deemed appropriate due to the bounded low degree of nodes in a molecular graph (≤ 4 in hydrocarbon structures).

Since subsequent match element selection beyond the initial depth zero refinement may involve multiple backtracking iterations, this particular implementation of the Ullman algorithm posts successively refined (marked for elimination) nodes to a diagonal storage structure of pointers of maximum dimensions, $\beta - 1 \times \beta - 1$. This avoids multiple dynamic memory allocation and deallocation during the course of the DFS. In the instance of a backtrack, the appropriate linked list in the diagonal storage structure is simply appended back onto the match structure. This process is repeated until the search has resulted in a success or a failure. Figure 6b depicts the storage structure and posted match nodes following the initial refinement procedure.

The initial refinement procedure is continued, examining every possible matching in the graph and removing invalid matchings, until a complete cycle through the matching structure without a removal has been achieved. Figure 6c displays the resultant, refined match structure at depth zero upon completion of the initial refinement. If the refinement at depth zero results in a null pointer in the anchor of the matching structure, the search is aborted.

Depth-First Search (DFS). Once the initial refinement has been completed, the DFS is implemented. DFS is a well-established concept in traversing undirected graphs.[54] In a typical DFS, a potential matching candidate that has not already been provisionally chosen is selected at a certain depth in the search. Then the depth is incremented, and the selection process is repeated. If an instance occurs where no potential matching exists at a certain depth, the depth is decremented, the match nodes removed during the previous refinement procedure are replaced in the match structure, and the next potential match is selected. In the Ullman approach, once a potential matching candidate has been selected during the DFS, the refinement procedure is then performed at each depth after each successive DFS selection.

A DFS procedure for the match structure following initial refinement (Figure 6c) would involve selecting match node (1,5), incrementing the depth counter, and then instituting a second refinement procedure. The second refinement procedure would differ from the previous in that any match node whose neighbors correspond to match nodes at depths less than the current depth must correspond to the one match node previously selected at each respective depth (e.g., (1,5)) during the DFS. This serves to increasingly constrain the potential permutations that survive subsequent refinement procedures. In this particular example, subsequent DFS and refinement will not result in any backtracking operations, and Figure 6d presents the resultant matching structure for the example query.

Exact Cover Procedure. From a graph theoretic perspective, a vertex (edge) of a subgraph is said to cover the vertexes (edges) in the query graph with which it is incident.[92] A cover is a set of vertexes (edges) whose union comprises the query graph.[60] An exact cover is a cover whereby each element of the query graph occurs exactly once in the cover set. An exact cover in the context of chemical structure disassembly is a procedure for elucidating a set of substructures whose union comprises a query molecule. This procedure is significant to GC methods of the structural increment type because many subfragment libraries contain structural ambiguities, hence multiple covers.

The exact cover complication, although well-known among general practitioners, has not been directly addressed in the literature. Several exact cover algorithms have been published[93-98] for arbitrary graphs, but the algorithm selected for use with MOSDAP is due to Frenz and Kreher.[99] A straightforward description of the updated procedure, pseudocode illustration of the algorithm, and example are presented by Kreher.[100] Although no direct comparisons with the other published algorithms were made, the Kreher algorithm was selected due to its simplicity, its efficiency, and its ability to determine all possible subfragment covers rather than the set that satisfies some assigned constraint. Typical constraints include covers resulting in the minimum quantity of subgraphs or the minimum weight whereby each subgraph is assigned a cumulative integer weight. The Kreher algorithm is an efficient backtracking operation with intermittent pruning.

To illustrate the cover procedure, Figure 7a presents a query structure, methyl acetoacetate, possessing multiple covers, and Figure 7b depicts the subset of substructures from the Hansen UNIFAC library[59] that are present in the query structure. Table 3 lists the groupings of substructures with their respective frequency of occurrence in each cover set. The estimated value of the target property (activity coefficient at infinite dilution at 25 °C) is also provided. Clearly, in the absence of measured data, it is not readily evident which calculated value is most appropriate.

The customary heuristic for such situations is to choose the cover resulting in the minimum number of subfragments, relying on the presumption that the larger substructures more accurately convey the cumulative effect on the property of

Table 3. Exact Cover Groupings for Figure 7

cover		substructures				γ^∞
1	i.d.	i	ii	iii		4.43
	quant	1	1	1		
2	i.d.	i	iv	v		34.4
	quant	1	1	1		
3	i.d.	ii	iv	vi		23.7
	quant	1	2	1		
4	i.d.	i	iv	vi	vii	15.6
	quant	1	1	1	1	

interest. This presumption has not been adequately addressed in the literature, although it is conversely possible that the smaller substructures, being much more prevalent within the fitted data set for the GC method, may more accurately characterize the property in certain instances. Another concern that can be addressed with the aid of the exact cover procedure is the inherent biasing of a structural library that contains ambiguous covers when the developer arbitrarily selects one cover set, ignoring the other potential candidates either intentionally or through ignorance of their existence. When these inherent biases in ambiguous substructure libraries are not adequately related to the practitioner, the method may be employed incorrectly.

While MOSDAP makes no provision for establishing the optimum cover for a particular implementation (as these heuristics may be highly empirical and method-specific), it does alert the user to the existence of multiple valid covers. The user must then determine the most appropriate structural breakdown. Several software programs dedicated to the quality analysis of estimated and actual data are available,[30,33,101,102] and MOSDAP has been developed primarily as a computational aid for these types of programs and for researchers in the field of chemical property estimation. MOSDAP may be implemented during the development phase to minimize structural ambiguity or postdevelopment to establish an appropriate set of ordering heuristics to help ameliorate the uncertainty of ambiguous structural libraries. The lexicographic notation adopted by us will permit the encoding of such rules or restrictions into the functional group libraries (Figure 3).

In addition to providing a mechanism for the determination of potential multiple covers for a particular query molecule, the exact cover procedure also addresses another limitation inherent in a sequential truncating substructure search. In a sequential search whereby each substructure is extracted from a query molecule in the order detected, the situation may arise where the extraction of a detected substructure results in a structural residual that is not capable of being partitioned into a subset of any of the remaining substructures, but the substructure file may still possess a valid cover for the query molecule. If a different substructure was extracted, it will result in a different structural residual which may be capable of being partitioned into a subset of the remaining substructures.

This situation is demonstrated with the example query molecule in Figure 8a. Figure 8d lists a subset of UNIFAC substructures that are contained within the query molecule of Figure 8a. It is evident by inspection that if the first substructure (i) was extracted from the query molecule, a structural residual (Figure 8b) would result that is incapable of being partitioned into a set of the remaining substructures. However, if substructure ii was initially selected for extraction, a structural residual (Figure 8c) results that is capable of being partitioned by the remaining substructures. This clearly illustrates that molecular structure disassembly is an inherently nonsequential operation for sets of substructures possessing structural ambiguity. Therefore, it cannot be assumed that a hierarchal ordering of substructures will allow for the determination of even a single structural disassembly for all potential queries.

Note that the exact cover search algorithm, when applied to many of the existing group contribution methods, may return degenerate covers for a given chemical. Symmetry within the query structure and subfragments possessing common structural moieties may lead to this degeneracy. Physically, each substructure may in fact cover positionally distinct locations within the molecule, but according to the simple frequency of occurrence accounting technique employed by most GC methods, the covers are considered degenerate. A graph theoretic exact cover procedure will interpret these as distinct, legal fragment breakdowns for the chemical even though a given GC method will treat the fragment lists as identical. To account for this effective degeneracy for a particular GC method, MOSDAP employs a simple degeneracy detection routine to cull degenerate covers.

COMPUTER IMPLEMENTATION

MOSDAP has been compiled as a 32-bit Windows 95/98 and as a UNIX-based application. The current myriad of programs based on substructure searching is fragmented, specialized, and difficult to assess, and MOSDAP was designed to operate as a peripheral search engine in an attempt to avoid being relegated to a specific program interface or implementation. In this client/server role, it may be employed in varying capacities by numerous applications involving structural dissassembly.

To illustrate the relative efficiences of the different search options supported by MOSDAP, the example structures present in Figure 9 were analyzed using the Hansen UNIFAC substructure library. The published library contains 108 substructures, but it was necessary to use 121 MOSDAP substructure declarations to represent the subfragment library. The substructures range in size from one to eight heavy atoms. Since MOSDAP uses an explicit connectivity representation, it is occasionally necessary to use multiple substructure declarations for a more general substructure representation for a particular GC method. For example, a disubstituted pyridine ring can be represented in multiple configurations.

Trial simulations were performed on a Hitachi M-Series laptop computer with a Pentium 120-MHz processor and 48 MB of RAM running under the Windows 98 operating system. The executable was compiled using Microsoft Visual C++ 5.0. Figure 9 presents the execution times resulting for each query structure using each of the three different search options: (1) sequential, nontruncating; (2) sequential, truncating; and (3) combinatorial, truncating. In the case of the combinatorial search (exact cover search), the number of resulting covers is also provided. In addition to the single-structure queries, a simulation was also performed on 1795 chemical structures from the DIPPR 911 database, ranging

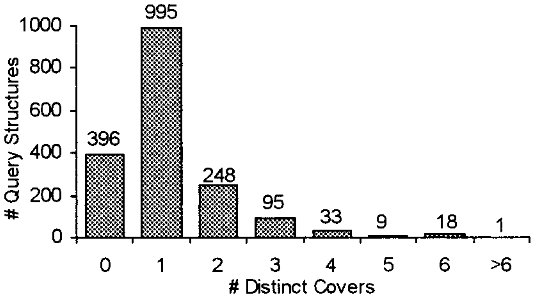


Figure 10. Multiple cover results (UNIFAC data set).

in size from 1 to 28 heavy atoms.[101] The resulting times for the file query are also provided. The results from the combinatorial search are presented in Figure 10. It is clear from the plot that 404 (29%) of the 1795 structures resulted in multiple covers, illustrating the potential of MOSDAP or a similar implementation to more adequately address the issue of estimated data-quality management.

STATUS OF THE MOSDAP SOFTWARE

We believe that the current version of MOSDAP is the most suitable program available for use in automating the development and implementation of GC methods. The substructure search algorithm is based upon recent advances by the chemical information community and provides a selection of search types depending upon the GC technique employed. The subfragment declaration syntax was designed to allow greater versatility than the preceding attempts for GC applications, including structurally and logically constrained subfragments and molecular feature declaration. Its novel implementation of the exact cover from graph theory to ascertain structural ambiguities provides a mechanism not previously available to those developing or using GC techniques. MOSDAP's ring perception algorithm is based upon the most recent published advances in the area and provides an efficient mechanism for obtaining ring information from a query structure.

The graph theoretic structural manipulation routines instituted within MOSDAP were coded so that the only constraint on query structure complexity is memory allocation. The size of a particular structure is theoretically limited by the number of heavy atoms (non-hydrogen) that can be represented by a signed 32-bit integer. Clearly, memory allocation and algorithmic efficiency will become an issue before a structure of this complexity can be achieved. A particular limitation of the structural manipulation routines within MOSDAP is the presumption that all structures can be represented by hydrogen-suppressed molecular graphs. This prevents the use of the truncating search operations for hydrogen-based subfragments which differentiate between other hydrogen atoms attached to the same heavy atom. In short, all substructures are assumed to be hydrogen-suppressed, and all hydrogens are treated identically. This limitation will be addressed in subsequent versions of the program.

Although it is believed that the implementation of advantageous graph theoretic algorithms, particulary the exact cover procedure, sets MOSDAP apart from other similar programs, one of the most advantageous attributes is its OOP representation of molecular structures. OOP allows MOSDAP to represent a molecular structure in greater detail and sophistication than traditional procedural programming techniques allow.[103] This greatly expands its potential for use in implementations beyond simple 2-D structural disassembly. Future versions of MOSDAP, through the implementation of inheritance and encapsulation, can be modified to accommodate stereochemistry, 3-D representation, nonprimary bonding, generic structures, etc., with minor modification to the existing code. Structural features and constraint operators can also be readily expanded to facilitate more specific and complex declarations. Additional structural manipulation routines such as maximum common subgraph (MCS)[26,27] and clique detection[100] can be added as member functions to the molecular structure class without modifying the data representation currently existing within MOSDAP.

The current version of MOSDAP supports the implementation of a large number of published and proprietary GC methods. The subfragment data files can be modified, or a new file can be created by the user, independently of MOSDAP. This versatility in subfragment declaration and search options allows a user the luxury of employing various GC methods to estimate the property of interest for the query molecule. This allows comparison of the estimated results between the various methods and/or experimental data with little effort from the user.

While MOSDAP is now primarily a platform to develop and implement simple two-dimensional structural increment type correlations, it is anticipated that future generations of the program will incorporate much more sophisticated features for the development of QSPRs. These will include a more articulate search syntax and structural topology member functions to allow MOSDAP to assess certain structural features internally so it is not reliant upon the complexity of the query structure syntax. In its present manifestation, the search syntax implemented in MOSDAP is still quite primitive. Currently, we are in the process of redesigning the search syntax to allow for the systematic addition of new search declarations and operators. The improved search syntax will add significant versatility by adopting the SMARTS substructure search syntax[104] developed by Daylight Chemical Services as well as employing additional search declarations specific to structural dissassembly searches.

Currently, MOSDAP serves as an augmenting function library to the DIPPR Project 911 ENVIRON 2001 software.[101] It precludes the need to store structural breakdowns for each GC method for each chemical within the PEARLS database. It also facilitates the updating of the GC methods by modifying the appropriate substructure representations in the subfragment library files so that it is not necessary to manually determine new groupings for each chemical within the database. This is especially important for users unfamiliar with the plethora of GC methods available for chemical property estimation. A new chemical may also be added to the database without first determining its respective subfragment groupings.

FURTHER INFORMATION REQUESTS

The research described in this paper responds to a specific need in chemical engineering to tie molecular representation and line notation to the requirements of structure-based physical property estimation. MOSDAP's open architecture and modular construction should permit interested researchers to test and compare additional substructure search and refinement strategies. Readers interested in obtaining a gratis copy of the MOSDAP C++ source code on a nondisclosure basis for research or academic use should contact author John Raymond at the address given at the beginning of this paper.

ACKNOWLEDGMENT

Development of the MOSDAP software was sponsored in part by the U.S. EPA Environmental Research Laboratory in Athens, GA, under Assistance Agreement No. CR823226-01-0, "Prediction of Chemical Parameters by Computer". Graduate Student Fellowship support of this work was provided by the National Institute of Standards and Technology (NIST), Physical and Chemical Properties Division, Chemical Science and Technology Laboratory (CSTL) in Boulder, CO. We express our appreciation to EPA Project Officer, Dr. Samuel Karickhoff, and NIST Project Officer, Dr. W. Mickey Haynes, for supporting and encouraging our efforts. Despite generous financial support for this work, the MOSDAP calculations and computer code have not been critically evaluated by NIST or the U.S. EPA and no official endorsement by either organization is implied. We extend our appreciation to the reviewers for their suggested improvements to this paper. The assistance of Dr. Donald Kreher of Michigan Technological University's Department of Mathematics in implementing the exact cover procedure is also gratefully acknowledged. On a personal note, author John Raymond dedicates this work "in memory of my beloved fiancé, Jodi Michelle Monosso, my one true love".

REFERENCES AND NOTES

(1) Chemical Abstract Services (CAS), Columbus, OH, 1998. <http://www.cas.org/>.

(2) Benson, S. W. *Thermochemical Kinetics*; Wiley: New York, 1968; Chapter 2.

(3) ChemDraw, Cambridge Soft Corporation, Cambridge, MA, 1998. <http://products.camsoft.com/chemdraw>.

(4) Howard, P. H.; Boethling, R. S.; Stiteler, W. M.; Meylan, W. M.; Hueber, A. E.; Beauman, J. A.; Larosche, M. E. Predictive Model for Aerobic Biodegradability Developed from a File of Evaluated Biodegradation Data. *Environ. Toxicol. Chem.* **1992**, *11*, 593−603.

(5) Lyman, W. J.; Reehl, W. F.; Rosenblatt, D. H. *Handbook of Chemical Property Estimation Methods*; American Chemical Society: Washington, DC, 1990.

(6) Reid, C. R.; Prausnitz, J. M.; Poling, B. E. *The Properties of Gases & Liquids*; McGraw-Hill: New York, 1987.

(7) Rogers, T. N.; Mullins, M. E.; Kline, A. *Environmental, Safety, and Health Data Estimation Technical Support Documents*; Project 912 Sponsor Release, July 1995; Design Institute for Physical Property Data (DIPPR); American Institute for Chemical Engineers (AICHE).

(8) Jochum, C.; Hicks, M. G.; Sunkel, J. *Physical Property Prediction in Organic Chemistry*; Springer-Verlag: Berlin, 1988.

(9) Baum, E. J. *Chemical Property Estimation: Theory and Application*; Lewis Publishers: Boca Raton, FL, 1998.

(10) Fredenslund, A.; Jones, R. L.; Prausnitz, J. M. Group-Contribution Estimation of Activity Coefficients in Nonideal Liquid Mixtures. *AICHE J.* **1975**, *21*, 1086−1099.

(11) Pintar, A. J. Estimation of Autoignition Temperature. Technical Support Document; Project 912, July 1996; Design Institute for Physical Property Data (DIPPR); American Institute for Chemical Engineers (AICHE).

(12) Brasie, W. C.; Liou, D. W. Estimating Physical Properties: Chemical Structure Coding. *Chem. Eng. Prog.* **1965**, *61*, 102−108.

(13) Jochelson, N.; Mohr, C. M.; Reid, R. C. The Automation of Structural Group Contribution Methods in the Estimation of Physical Properties. *J. Chem. Doc.* **1968**, *8*, 113−122.

(14) Smith, E. G. *Wiswesser Line-Formula Chemical Notation Methods;* McGraw-Hill: New York, 1968.

(15) Sussenguth, E. H. A Graph-Theoretic Algorithm for Matching Chemical Structures. *J. Chem. Doc*. **1965**, *6*, 36−43.

(16) Barnard, J. M. Substructure Searching Methods: Old & New. *J. Chem. Inf. Comput. Sci*. **1993**, *33*, 532−538.

(17) Ming, T.; Tauber, S. J. Chemical Structure and Substructure Search by Set Reduction. *J. Chem. Doc*. **1971**, *11*, 47−51.

(18) Adams, J. T.; So, E. M. Automation of Group-Contribution Techniques for Estimation of Thermophysical Properties. *Comput. Comput. Eng.* **1985**, *9*, 269−284.

(19) Qu, D.; Su, J.; Muraki, M.; Hayakawa, T. A Decoding System for a Group Contribution Method. *J. Chem. Inf. Comput. Sci*. **1992**, *32*, 448−452.

(20) Drefahl, A.; Reinhard, M. Similarity-Based and Evaluation of Environmentally Relevant Properties for Organic Compounds in Combination with the Group Contribution Approach. *J. Chem. Inf. Sci.* **1993**, *33*, 886−895.

(21) Figueras, J. Substructure Search by Set Reduction. *J. Chem. Doc.* **1972**, *12*, 237−246.

(22) Qu, D.; Fu, B.; Muraki, M.; Hayakawa, T. An Encoding System for a Group Contribution Method. *J. Chem. Inf. Comput. Sci.* **1992**, *32*, 443−447.

(23) Weinenger, D. SMILES, a Chemical Language and Information System. 1. Introduction to Methodology and Encoding Rules. *J. Chem. Inf. Comput. Sci.* **1988**, *28*, 31−36.

(24) Weinenger, D.; Weinenger, A.; Weinenger, J. L. SMILES. 2. Algorithm for Generation of Unique SMILES Notation. *J. Chem. Inf. Comput. Sci.* **1989**, *29*, 97−101.

(25) Smiles Tutorial, Daylight Chemical Services, 1998. <http://www.daylight.com/dayhtml/smiles~intro.htm>.

(26) Willett, P.; Winterman, V. A. Algorithms for the Calculation of Similarity in Chemical Structure Databases. In *Concepts and Applications of Molecular Similarity*; Johnson, M. A., Maggiore, G. M., Eds.; Wiley: New York, 1990.

(27) Brint, A. T.; Willett, P. Algorithms for the Identification of Three-Dimensional Maximal Common Substructures. *J. Chem. Inf. Comput. Sci.* **1987**, *27*, 152−158.

(28) PC UNIFAC, BrisSoftware, Atlanta, GA.

(29) Estimation Program Interface, Syracuse Research Corporation, North Syracuse, NY, 1998. <http://esc.syrres.com/~esc1/estsoft.htm>.

(30) Cranium, Molecular Knowledge Systems; New Bedford, NH, 1998. <http://www.molknow.com/cranium.htm>.

(31) GCDATA, ProSim, Toulouse, France, 1996.

(32) GC-Estimate, Novel Advanced Systems Corp; Ann Arbor, MI.

(33) PREDICT, Dragon Technology, Inc.; Golden, CO, 1998. <http://www.mwsoftware.com/dragon>.

(34) Linert, W.; Margl, P.; Nusterer, E. The Use of Enhanced Operator-Machine Interfaces in Computer Aided Molecular Design. *Comput. Chem.* **1991**, *15*, 1−10.

(35) Muller, C.; Scacchi, G.; Come, G. A Compiler for a Linear Chemical Notation. *Computers Chem*. **1991**, *15*, 337−342.

(36) Leung, K.; Chau, F.; Kwok, P.; Lau, W. ChemISTools: A Computer Software for Chemical Information Systems. *Comput. Chem.* **1997**, *21*, 161−166.

(37) Ash, J. E.; Hyde, E. *Chemical Information System*; Ellis Horwood: New York, 1975.

(38) Davis, C. H.; Rush, J. E. *Information Retrieval and Documentation in Chemistry*; Greenwood Press: London, 1974.

(39) Ash, S.; Cline, M. A.; Homer, R. W.; Hurst, T.; Smith, G. B. SYBYL Line Notation (SLN): A Versatile Language for Chemical Structure Representation. *J. Chem. Inf. Comput. Sci.* **1997**, *37*, 71−79.

(40) Tonnelier, C. A.; Fox, J.; Judson, P.; Krause, P.; Pappas, N.; Patel, M. Representation of Chemical Structures in Knowledge-Based Systems: The StAR System. *J. Chem. Inf. Comput. Sci*. **1997**, *37*, 117−123.

(41) Karabunarliev, S.; Ivanov, J.; Mekenyan, O. Coding of Chemical Structures Based on a Line Notation. *Comput. Chem.* **1994**, *18*, 189−193.

(42) Franklin, J. L. Prediction of Heat and Free Energies of Organic Compounds. *Ind. Eng. Chem.* **1949**, *41*, 1070−1076.

(43) Franklin, J. L. Calculation of the Heats of Formation of Gaseous Free Radicals. *J. Chem. Phys.* **1953**, *21*, 2029−2033.

(44) Joback, K. *A Unified Approach to Physical Property Estimation Using Multivariate Statisitical Techniques*. MS Thesis, Massachesetts Institute of Technology (MIT), Cambridge, MA, June 1982.

(45) Database of SMILES Notations, Syracuse Research Corporation, North Syracuse, NY, 1998. <http://esc.syrres.com/~esc1/database.htm>.

(46) Feldman, A.; Hodes, L. An Efficient Design for Chemical Structure Searching. I. The Screens. *J. Chem. Inf. Comput. Sci*. **1975**, *15*, 147−152.

(47) Downs, G. M.; Barnard, J. M. Techniques for Generating Descriptive Fingerprints in Combinatorial Libraries. *J. Chem. Inf. Comput. Sci.* **1997**, *37*, 59−61.

(48) Akutsu, T.; Bao, F. Approximating Minimum Keys and Optimal Substructure Screens. Computing and Combinatorics. Second Annual International Conference (COCOON), Hong Kong, June 1996.

(49) Willett, P. A Screen Set Generation Algorithm. *J. Chem. Inf. Comput. Sci.* **1979**, *19*, 159−162.

(50) Adamson, G. W.; Clinch, V. A.; Creasey, S. E.; Lynch, M. F. Distributions of Fragment Representations in a Chemical Substructure Search Screening System. *J. Chem. Doc.* **1974**, *14*, 72−74.

(51) Adamson, G. W.; Bush, J. A.; McLure, A.; Lynch, M. F. An Evaluation of a Substructure Search Screen System Based on Bond-Centered Fragments. *J. Chem. Doc.* **1974**, *14*, 44−48.

(52) Adamson, G. W.; Lambourne, D. R.; Lynch, M. F. Analysis of Structural Characteristics of Chemical Compounds in a Large Computer-Based File. Part III. Statistical Association of Fragment Incidence. *J. Chem. Soc., Perkin Trans. 1* **1972**, 2428−2433.

(53) Welford, S. M.; Lynch, M. F.; Barnard, J. M. Computer Storage and Retrieval of Generic Chemical Structures in Patents.5.Algorithmic Generation of Fragment Descriptors for Generic Structure Screening. *J. Chem. Inf. Comput. Sci.* **1984**, *24*, 57−66.

(54) Tremblay, J. P.; Sorenson, P. G. *Introduction to Data Structures with Applications*; McGraw-Hill: New York, 1984.

(55) Figueras, J. Ring Perception Using Breadth-First Search. *J. Chem. Inf. Comput. Sci.* **1996**, *36*, 986−991.

(56) Bauerschmidt, S.; Gasteiger, J. Overcoming the Limitations of a Connection Table Description: A Universal Representation of Chemical Species. *J. Chem. Inf. Comput. Sci.* **1997**, *37*, 705−714.

(57) Willett, P. A Review of Chemical Structure Retrieval System. *J. Chemometrics* **1987**, *1*, 139−155.

(58) Gaurilov, D.; Vinogradov, O. Object-Oriented Programming and Object-Oriented Simulation. Proceedings of the SCSC, Ottawa, Canada, 1995.

(59) Hansen, H. K. Vapor−Liquid Equilibria by UNIFAC Group Contribution. 5. Revision and Extension. *Ind. Eng. Chem. Res.* **1991**, *10*, 2352−2355.

(60) Read, R. C. *Graph Theory and Computing*; Academic Press: New York, 1972; pp 267−283.

(61) Downs, G.; Gillet, V.; Holliday, J.; Lynch, M. Theoretical Aspects of Ring Perception and Development of the Extended Set of Smallest Rings Concept. *J. Chem. Inf. Comput. Sci.* **1989**, *29*, 187−206.

(62) Downs, G.; Gillet, V.; Holliday, J.; Lynch, M. Review of Ring Perception Algorithms for Chemical Graphs. *J. Chem. Inf. Comput. Sci.* **1989**, *29*, 172−187.

(63) Fan, T.; Panaye, A.; Doucet, J.; Barbu, A. Ring Perception. A New Algorithm for Directly Finding the Smallest Set of Smallest Rings from a Connection Table. *J. Chem. Inf. Comput. Sci.* **1993**, *33*, 657−662.

(64) Balducci, R.; Pearlman, R. Efficient Exact Solution of the Ring Perception Problem. *J. Chem. Inf. Comput. Sci.* **1994**, *34*, 822−831.

(65) Stobaugh, R. E. Chemical Substructure Searching. *J. Chem. Inf. Comput. Sci.* **1985**, *25*, 71−275.

(66) Barnard, J. M. Problems of Substructure Search and Their Solution. In *Chemical Structures*; Warr, W., Ed.; Springer-Verlag: Berlin, 1988.

(67) Willett, P. Processing of Three-Dimensional Chemical Structure Information Using Graph-Theoretic Techniques. *Online Information 90*; London, 1990; pp 115−127.

(68) Brown, R. D.; Jones, G.; Willett, P. Matching Two-Dimensional Chemical Graphs Using Genetic Algorithms. *J. Chem. Inf. Comput. Sci.* **1994**, *34*, 63−70.

(69) Dengler, A.; Ugi, I. A Central Atom Based Algorithm and Computer Program for Substructure Search. *Comput. Chem.* **1991**, *15*, 103−107.

(70) Jun, X.; Maosen, Z. HBA: New Algorithm for Structural Match and Applications. *Tetrahedron Comput. Methodol.* **1989**, *2*, 75−83.

(71) Von Scholley, A. A Relaxation Algorithm for Generic Chemical Structure Searching. *J. Chem. Inf. Comput. Sci.* **1984**, *24*, 235−241.

(72) Wang, Y. K.; Fan, K. C. Adaptive Optimization for Solving a Class of Subgraph Isomorphism Problems. *IEEE* **1995**, 44−49.

(73) Ozawa, K.; Yasud, T.; Fujita, S. Substructure Search with Tree-Structured Data. *J. Chem. Inf. Comput. Sci.* **1997**, *37*, 688−695.

(74) Xiao, Y.; Qiao, Y.; Zhang, J.; Lin, S.; Zhang, W. A Method for Substructure Search by Atom-Centered Multilayer Code. *J. Chem. Inf. Comput. Sci.* **1997**, *37*, 701.

(75) Lesk, A. Detection of 3-D Patterns of Atoms in Chemical Structures. *Commun. ACM* **1979**, *22*, 219−224.

(76) Ullman, J. R. An Algorithm for Subgraph Isomorphism. *J. Assoc. Comput. Mach.* **1976**, *23*, 31−42.

(77) Brint, A.; Mitchell, E.; Willet, P. Substructure Searching in Files of Three of Dimensional Chemical Structures. In *Chemical Structures*; Warr, W., Ed.; Springer-Verlag: Berlin, 1988.

(78) Artymiuk, P. J.; MacKenzie, A. B.; Grindley, H. M.; Poirette, A. R.; Rice, D. W.; Ujah, E. C.; Willet, P. Representation and Searching of Three-Dimensional Protein Structures Using Graph Theory: Part 5. *Chem. Des. Aut. News* **1994**, 6−17.

(79) Sheridan, R. P.; Nilakantan, R.; Rusinko, A.; Bauman, N.; Haraki, K. S.; Venkataraghavan, R. 3DSEARCH: A System for Three-Dimensional Substructure Searching. *J. Chem. Inf. Comput. Sci.* **1989**, *29*, 255−260.

(80) Wang, T.; Zhou, J. 3DFS: A New 3D Flexible Searching System for Use in Drug Design. *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 71−77.

(81) Clark, D. E.; Murray, C. W. PRO_LIGAND: An Approach to de Novo Molecular Design. 5. Tools for the Analysis of Generated Structures. *J. Chem. Inf. Comput. Sci.* **1995**, *35*, 914−923.

(82) Brint, A. T.; Willett, P. Pharmacophoric Pattern Matching in Files of 3D Chemical Structures: Comparison of Geometric Searching Algorithms. *J. Mol. Graph.* **1987**, *5*, 49−56.

(83) Downs, G. M.; Lynch, M. F.; Willett, P.; Manson, G. A.; Wilson, G. A. Transputer Implementations of Chemical Substructure Searching Algorithms. *Tetrahedron Comput. Methodol.* **1988**, *1*, 207−217.

(84) Holliday, J. D.; Lynch, M. F. Computer Storage and Retrieval of Generic Chemical Structures in Patents. 16. The Refined Search: An Algorithm for Matching Components of Generic Chemical Structures at the Atom-Bond Level. *J. Chem. Inf. Comput. Sci.* **1995**, *35*, 1−7.

(85) Mitchell, E. M.; Artymiuk, P. J.; Rice, D. W.; Willett, P. Use of Techniques Derived from Graph Theory to Compare Secondary Structure Motifs in Proteins. *J. Mol. Biol.* **1989**, *212*, 151−166.

(86) Artymiuk, P. J.; Grindley, H. M.; Poirette, A. R.; Rice, D. W.; Ujah, E. C.; Willett, P. Identification of β-Sheet Motifs, of Ψ-Loops, and of Patterns of Amino Acid Residues in Three-Dimensional Protein Structures Using a Subgraph-Isomorphism Algorithm. *J. Chem. Inf. Sci.* **1994**, *34*, 54−62.

(87) Willett, P.; Wilson, T. Atom-by-Atom Searching Using Massive Parallelism. Implementation of the Ullman Subgraph Isomorphism Algorithm on the Distributed Array Processor. *J. Chem. Inf. Sci.* **1991**, *31*, 225−233.

(88) Bone, R. G.; Villar, H. O. Exhaustive Enumeration of Molecular Substructures. *J. Comput. Chem.* **1997**, *18*, 86−107.

(89) Hurst, T. Flexible 3D Searching: The Directed Tweak Technique. *J. Chem. Inf. Comput. Sci.* **1994**, *34*, 190−196.

(90) *Mathematical Challenges from Theoretical/Computational Chemistry;* National Research Council; National Academy Press: Washington, DC, 1995.

(91) Swamy, M. N. S.; Thulasiraman, K. *Graphs, Networks, and Algorithms*; Wiley: New York, 1981; pp 536−538.

(92) Foulds, L. R. *Graph Theory Applications*; Springer-Verlag: Berlin, 1992; pp 123−124.

(93) Balas, E.; Samuelson, H. A Node Covering Algorithm. *Naval Res. Log. Quart.* **1977**, *24*, 213−233.

(94) Harche, F.; Thompson, G. The Column Subtraction Algorithm: An Exact Method for Solving Weighted Set Covering, Packing, and Partitioning Problems. *Comput. Ops. Res.* **1994**, *21*, 689−705.

(95) Beasely, J. An Algorithm for Set Covering Problem. *Eur. J. Oper. Res.* **1987**, *31*, 85−93.

(96) Beasely, J.; Jornsten, K. Enhancing an Algorithm for Set Covering Problems. *Eur. J. Oper. Res.* **1992**, *58*, 293−300.

(97) Bertolazzi, P.; Sassano, A. An O(mn) Algorithm for Regular Set-Covering Problems. *Theor. Comput. Sci.* **1987**, *54*, 237−247.

(98) Baker, E. Efficient Heuristic Algorithms for the Weighted Set Covering Problem. *Comput. Oper. Res.* **1981**, *8*, 303−310.

(99) Frenz, T. C.; Kreher, D. L. An Algorithm for Enumerating Distinct Cyclic Steiner Systems. *J. Combin. Math. Combin. Comput.* **1992**, *11*, 23−32.

(100) Kreher, D. L.; Stinson, D. R. *Combinatorial Algorithms: Generation, Enumeration and Search*. CRC Press: Boca Raton, FL, 1998.

(101) Rogers, T.; Kline, A.; Miller, M.; Wieber, D.; Mullins, M. *ENVIRON 2001, Environmental, Safety, and Health Data and Estimates for the 21st Century*; Project 911, Sponsor Release, March 1999; Design Institute for Physical Property Data (DIPPR); American Institute of Chemical Engineers (AICHE): New York.

(102) Data Expert, Epcon International, Houston, TX, 1998. <http://www.epcon.com>.

(103) Mavrovouniotis, M. L.; Pricket, S.; Constantinov, L. Object-Oriented Estimation of Properties from Molecular Structure. European Symposium on Computer Aided Process Engineering-1, S353−S360.

(104) SMARTS, Daylight Chemical Services. 1998. <http://www.daylight.com/products/smarts_kit.html>.

CI9803334