Chem. Inf. Comput. Sci. **1988**, *28*, 31-36.

(15) Gund, P.; Wipke, W. T.; Langridge, R. Computer searching of a molecular structure file for pharmacophoric patterns. In *Proceedings of the International Conference on Computers in Chemical Research and Education*, Ljubljana; Elsevier: Amsterdam, 1974; Vol. 3, pp 5/33-38.

(16) Gund, P. Three-dimensional pharmacophoric pattern searching. *Prog. Mol. Subcell. Biol.* **1977**, *5*, 117-143.

(17) Jakes, S. E.; Willett, P. Pharmacophoric pattern matching in files of 3-D chemical structures: selection of interatomic distance screens. *J. Mol. Graphics* **1986**, *4*, 12-20.

(18) Jakes, S. E.; Watts, N.; Willett, P.; Bawden, D.; Fisher, J. D. Pharmacophoric pattern matching in files of 3D chemical structures: evaluation of search performance. *J. Mol. Graphics* **1987**, *5*, 41-48.

(19) Brint, A. T.; Willett, P. Pharmacophore pattern matching in files of 3D chemical structures: comparison of geometric searching algorithms. *J. Mol. Graphics* **1987**, *5*, 49-56.

(20) Behling, R. W.; Yamane, T.; Navon, G.; Jelinski, L. W. Conformation of acetylcholine bound to the nicotinic acetylcholine receptor. *Proc. Natl. Acad. Sci. U.S.A.* **1988**, *85*, 6721-6725.

(21) Schulman, J. M.; Sabio, M. L.; Disch, R. L. Recognition of Cholinergic Agonists by the Muscarinic Receptor. 1. Acetylcholine and Other Agonists with the NCCOCC Backbone. *J. Med. Chem.* **1983**, *26*, 817-23.

(22) Tollenaere, J. P. Muscarinic pharmacophore identification. *Trends Pharmacol. Sci.* **1984**, *5*, 85-86.

(23) Dolata, D. P.; Leach, A. R.; Prout, K. WIZARD: AI in conformational analysis. *J. Comput.-Aided Mol. Des.* **1987**, *1*, 73-85.

(24) Wipke, W. T.; Hahn, M. A. Analogy and Intelligence in Model Building. In *Artifical Intelligence Applications in Chemistry*; Pierce, T. H., Hohne, B. A., Eds.; ACS Symposium Series 306, American Chemical Society: Washington, DC, 1986; pp 136-146. Hahn, M.; Wipke, W. T. Poster session: analogy and intelligence in model building (AIMB). In *Chemical Structures*; Warr, W. A., Ed.; Springer-Verlag, Berlin, 1988; pp 267-278.

(25) Wenger, J. C.; Smith, D. H. Deriving Three-Dimensional Representations of Molecular Structure from Connection Tables Augmented with Configuration Designations Using Distance Geometry. *J. Chem. Inf. Comput. Sci.* **1982**, *22*, 29-34.

(26) Martin, Y.; Danaher, E. B.; May, C. S.; Weininger, D. MENTHOR, a database system for the storage and retrieval of three-dimensional molecular structures and associated data searchable by substructural, biologic, physical, or geometric properties. *J. Comput.-Aided Mol. Des.* **1988**, *2*, 15-29.

(27) Sheridan, R. P.; Venkataraghavan, R. Designing novel nicotinic agonists by searching a database of molecular shapes. *J. Comput.-Aided Drug Des.* **1987**, *1*, 243-256.

(28) DesJarlais, R. L.; Sheridan, R. P.; Seibel, G. L.; Dixon, J. S.; Kuntz, I. D.; Venkataraghavan, R. Using Shape Complementarity as an Initial Screen in Designing Ligands for a Receptor Binding Site of Known Three-Dimensional Structure. *J. Med. Chem.* **1988**, *31*, 722-729.

3DSEARCH: A System for Three-Dimensional Substructure Searching

ROBERT P. SHERIDAN, RAMASWAMY NILAKANTAN, ANDREW RUSINKO III,
NORMAN BAUMAN, KEVIN S. HARAKI, and R. VENKATARAGHAVAN*

Medical Research Division, Lederle Laboratories, American Cyanamid Company,
Pearl River, New York 10965

Received May 5, 1989

The system 3DSEARCH is used to search for three-dimensional substructures (for example, pharmacophores) in databases of coordinates. Searches are divided into two parts, a fast prescreen using an inverted key system and a slower atom-by-atom geometric search using the algorithm described by Ullman (*J. Assoc. Comput. Mach.* **1976**, *23*, 31-42). Features to handle angle/dihedral constraints and to take into account "excluded volume" are implemented as part of the geometric search. With this strategy, searches of typically sized queries over large databases (>200 000 entries) take only a few minutes. The speed of the system is demonstrated with a few examples of queries derived from pharmacophores in the literature.

INTRODUCTION

Now that three-dimensional molecular modeling is becoming widely used, "pharmacophore" models for a variety of biological activities are appearing in the literature in increasing numbers. A pharmacophore is a spatial arrangement of chemical groups (usually atoms), common to all active molecules, that is recognized by a single receptor. Pharmacophores can be deduced by a variety of techniques.[1-3] The specifications of the pharmacophores in the literature generally have the following properties: (1) The relationship between the atoms is described by distances and/or angles rather than in terms of bonds. (2) The atoms might be described by chemical property (cation, H-bond donor, etc.) rather than by element type. Also, an atom might be a "dummy", a point that is used to define a geometry, but which takes up no volume (for example, the centroid of a ring).

Often, we want to find which molecules in a set, say, a database of proprietary compounds, contain a particular pharmacophore. To do this we need to have database(s) of coordinates and a method of searching them. In this paper we describe 3DSEARCH, a system to define and search for three-dimensional substructures. The database to be searched may be generated from experimental coordinates (e.g., the Cambridge Structural Database[4]) or from connection tables as discussed in the previous paper.[5]

REPRESENTATION OF ATOM TYPES

For each structure in our database we list a unique identifying integer and the number of non-hydrogen atoms. For each (non-hydrogen) atom are listed the atom type and *xyz* coordinates. As discussed in the previous paper,[5] the atom type consists of five fields: element (He-U); number of non-hydrogen neighbors, i.e., bonded atoms (0-8); the number of π electrons (0-2); the expected number of attached hydrogens (0-4); formal charge (-1, 0, 1). Four types of dummy atoms are used to define geometric points in space. "Element" types D5 and D6 are the centroids of planar 5- and 6-membered rings. DP are along the perpendicular to these rings. DL are connected to each heteroatom and aligned along the sum of vectors from its neighbors.

DEFINITION OF QUERIES

In general, a query is a question formulated to interrogate a database. In our case, a "query" is a three-dimensional substructure consisting of a set of atoms and a description of their spatial relationship. A structure in the database is a "hit" if and only if it contains the query.

Definition of Spatial Relationship between Atoms. The most basic representation of a spatial relationship is as a set of lower and upper bounds to the interatomic distances. A simple example of a structure that contains a query is shown in Figure

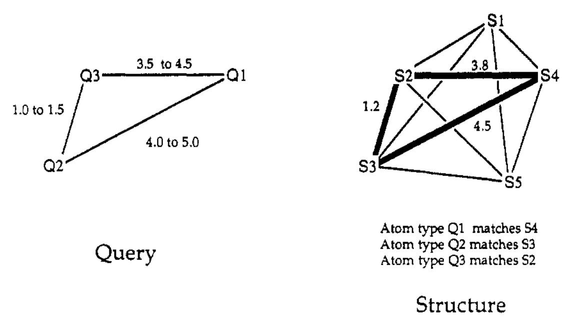


Figure 1. Example of a structure that contains a query. The points Q1-3 are query atoms, and S1-5 are structure atoms. In this example, we show that the structure contains the query since a subset of structure atoms (S4, S3, S2) matches the query atoms (P1, P2, P3) in type and the interatomic distances are within the range specified in the query.

1. In this case we imagine that Q1 matches S4, Q2 matches S3, and Q3 matches S2 in atom type, and the interatomic distances between those structure atoms are within the corresponding distance bounds in the query.

Besides specifying distance bounds, a query can also specify that three or four of its atoms must form, respectively, an angle or dihedral angle within lower and upper bounds. Sometimes one needs to specify "excluded volumes" in a query, volumes that are not allowed to be occupied by any atom in a structure. To define an excluded volume we prepare a file containing a "model", which consists of a set of "pattern" points and a set of "outrigger" points, each point with fixed *xyz* coordinates. A specified radius, defining a spherical volume to be excluded, can be associated with each of the outriggers. Of course, for the model to be meaningful in relationship to the query, the pattern points must be in one-to-one correspondence with the query atoms and the distances between pattern atoms must be compatible with the lower and upper bounds in the query.

Matches in Atom Type. Each atom in the query is specified by the same five-field atom type used in the database except for two features: (1) One may use "wild cards" (Wi) in any of the fields. (2) One may specify more than one atom type per query atom. For instance, we could allow an atom to be any of six possible types of cations

type	element	neighbors	π	H's	chg
1	N	1	0	3	1
2	N	2	0	2	1
3	N	3	0	1	1
4	N	3	1	0	1
5	N	4	0	0	1
6	S	3	0	0	1

or we could define a generic cation

1	Wi	Wi	Wi	Wi	1

Query and structure atoms match if all of the fields in the structure atom are identical with the corresponding fields of at least one of the types allowed for the query atom. For instance, the following is a match:

	type	element	neighbors	π	H's	chg
Q1	1	O	2	0	0	0
	2	S	2	0	0	0
	3	N	2	1	0	0
S4		S	2	0	0	0

Also, any value in a field for a structure atom is allowed when the corresponding value in a query atom type is a wild card, as in the following match:

	type	element	neighbors	π	H's	chg
Q1	1	N	Wi	Wi	Wi	1
S4		N	2	1	0	1

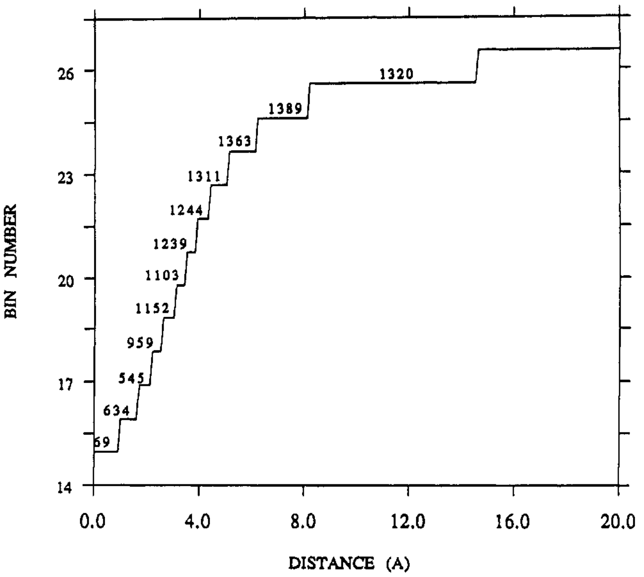


Figure 2. Distance bins for constructing keys. The size of the bins is determined by the formula: bin number = 5 arctan $((D - 3.0)/2)$ + 20.0. Indicated on each bin is the number of keys with the corresponding distance range in the 223 988 structures in the three-dimensional CL File.

In the current implementation, dummy atoms match only with dummy atoms and "real" atoms with real atoms, even when the element field for a query atom is a wild card. For this reason no query atom is allowed to have both real and dummy atom types.

SEARCH STRATEGY

We have adopted the strategy used by others wherein a search is divided into two independent phases. The first phase, a "key search", is a rapid prescreen to eliminate those structures in the database that cannot possibly contain the query. It is meant to reduce the number of structures to a few percent of the initial list. The second phase is a slower atom-by-atom geometric search on the set of structures that are hits in the key search.

Preparation of Keys. Our implementation of the key search is based on resolving each molecule in a database into a set of constituent descriptors or "keys". A key search finds structures that contain all the keys contained in the query. Here, we use an "inverted" key scheme, in which each key in the database is associated with a list of structures that contain it (in contrast to a "direct" key scheme where each structure is associated with the list of keys it contains). Our keys have the form

atom type 1 – {distance} – atom type 2

where the atom type includes all five fields. The through-space distance between any two atoms is divided into a number of discrete "bins", so that {distance} is expressed as a bin number. The effectiveness of the keys will be maximal when all the keys occur in the database at about the same frequency. Analysis of the distribution of interatomic distances in three-dimensional databases shows a peak in the 3–5-Å range. Thus, to achieve a more even distribution, we use narrow bins in this range and progressively wider bins in the flanking regions. Empirically we find that the formula

$$\text{distance bin number} = 5 \arctan ((D - 3.0)/2) + 20.0$$

works well. Figure 2 is a plot of the bin number obtained by using the above formula versus the interatomic distance D.

A key can be represented as an integer (the bin number) plus a bitmap consisting of 10 fields of bits (2 atoms, 5 fields per atom). Within each field, each bit represents a possible value for the corresponding property. For example, if the first atom in the key were a nitrogen, the seventh bit (i.e., atomic number = 7) in its element field would be set to 1. A key can be canonicalized so that its definition is not sensitive to the order of the atoms in the pair.

To generate a set of unique keys and their list of associated structures from a three-dimensional database, we follow the procedure

```
For structure i in the database:
  Read the atom types and coordinates.
  For each pair of atoms in structure i:
    Define a candidate key for this pair and canonicalize
    it.
    If the candidate key is identical with key k, which is
    already in the list of unique keys, add the identifier
    for structure i to the list associated with key k.
    Otherwise, add the candidate key to the list of unique
    keys and add the identifier of structure i to the list
    of structures associated with this new key.
  End.
End.
```

From this procedure we generate two files, a sequential file of unique keys and an indexed file containing the list of the structures associated with each key. Each list is in the form of a bitmap in which the *i*th bit is set to 1 if structure *i* contains the key. The three-dimensional CL File (the set of Cyanamid proprietary compounds), which was generated from the connection tables as described in the previous paper,[5] consists of 223 988 structures in which there are 13 360 unique keys.

Execution of Key Search. A key search is executed by manipulating lists of structures represented as bitmaps:

```
Read in the list of unique keys.
Initialize a bitmap A so that the ith bit is set to 1 if the
database contains structure i.
For every pair of query atoms for which a distance range
has been specified:
  Initialize a bitmap B with all bits set to 0.
  Create and canonicalize a key "mask" by using the
  allowed atom types and the distance range (see
  next paragraph) for that pair of query atoms.
  For every key k in the key list:
    If key k is a match of the key mask in atom types
    and distance (see next paragraph), extract the
    bitmap C containing the list of structures associated
    with key k.
    Perform the bitwise logical operation B ← B <OR>
    C.
  End.
  Perform the bitwise logical operation A ← A <AND>
  B.
End.
```

The final bitmap **A** represents the list of structures in the database that contain all the pairs of atoms (with correct types and distances) contained in the query.

A key mask has the same representation as a key. When we generate a mask for a pair of atoms in the query, we account for multiple atom types by setting to 1, in each field, a bit for each of the allowed values. A wild card in any field means that *all* the bits corresponding to that field are set to 1. For example, imagine the pair of query atoms with the distance range 3.5–5.0 Å:

atom	type	element	neighbors	π	H's	chg
Q1	1	N	4	0	0	1
	2	N	3	0	1	1
	3	S	3	0	0	1
Q2	1	O	1	0	1	0
	2	S	1	0	Wi	Wi

In the mask, the bits corresponding to N and S would be set to 1 in the element field of Q1, all the bits in the chg field of Q2 would be set to 1, etc.

The key corresponding to

	element	neighbors	π	H's	chg
atom 1	N	3	0	1	1
atom 2	O	1	0	1	0
distance bin	22				

would be considered a match for the mask since for each of the 10 fields there is a corresponding bit set to 1 in both the key and the mask and the distance bin falls within the range of bin numbers associated with the upper and lower bounds (3.5–5.0 Å → bins 21–23).

By setting bits in this way we make handling wild cards straightforward, simplify the loop structure of the key search algorithm, and thereby increase the speed. However, the definition of an atom may become ambiguous, occasionally leading to spurious matches. For instance, the key

	element	neighbors	π	H's	chg
atom 1	N	3	0	1	1
atom 2	O	1	0	0	–1
distance bin	22				

would be a match for the mask, even though the specification of atom 2 is not allowed by the query. Such spurious matches are eliminated in the subsequent geometric search.

Geometric Search. Although the key search finds structures that contain pairs of atoms with the correct types at roughly the correct distances, the pairs may or may not be interrelated as defined by the query. Thus, we need an atom-by-atom geometric search.

In practice, we store the three-dimensional database as an indexed file so that any particular structure can be accessed directly by its structure identifier. Our usual procedure is to go through a list of structure identifiers (such as that produced by the key search). For each structure, we access the atom types and *xyz* coordinates, calculate the interatomic distances, and proceed with the geometric search. In our current implementation, we may restrict the geometric search to those structures for which the number of non-hydrogen, non-dummy atoms falls within a specified range.

We use the isomorphous subgraph algorithm of Ullman,[6] which has been reviewed by Brint and Willett[7] and shown by them to be more efficient than competing algorithms. The algorithm starts by constructing a correspondence matrix **M**. If the atom type of query atom *i* matches the type of structure atom *j* (thus *i* might correspond to *j*), then $M(i,j) = 1$; otherwise, $M(i,j) = 0$. The algorithm then does a backtrack search (modifying **M** as it proceeds and checking for self-consistency) looking for a *set* of *i*–*j* pairs such that *each* query atom is paired with exactly one distinct atom in the structure and the interatomic distances of the structure fall within the bounds of the corresponding distances in the query. Once such a set is found, the query is shown to be contained in the structure. (The speed of the algorithm comes from the property that, if there is no acceptable set of one-to-one pairs, this is usually discovered early in the backtrack search.) In the unmodified algorithm, a structure would be declared a hit as soon as the first set of one-to-one pairs is discovered. In our implementation (as described in the following two paragraphs), the set is also tested for angle/dihedral constraints and for excluded volume. If the set passes these tests, the structure is declared a hit; otherwise, the backtrack algorithm "backs up" one level and continues its search for an alternative set.

An example of an angle/dihedral test during the backtrack search can be discussed in reference to Figure 1. Assume we

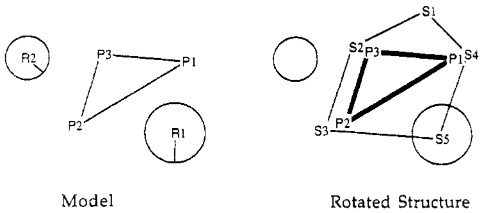


Figure 3. Example of a structure rotated/translated onto a model. The atoms P1–3 represent pattern points in three-dimensional space that correspond to query points Q1–3. Atoms R1 and R2 are outriggers with associated radii. They define an excluded volume. In this example, we assume the same correspondence of structure atoms to query atoms as in Figure 1. Structure atom S5 falls into the excluded volume defined by R1.

specified in the query that the angle Q1–Q3–Q2 was to be within, say, 100–150°. The structure (which contains the query) would be a hit only if the angle formed by the matched atoms in the structure (S4–S2–S3) fell within that range. In our current implementation, we specify dihedral angle ranges as absolute values. This is because we are often interested in structures containing a given dihedral angle *or* that structure's mirror reflection, which contains the dihedral angle of the opposite sign. If there is more than one angle/dihedral constraint, all must be satisfied.

An example of how a structure is tested against a predefined excluded volume during the backtrack search is shown in Figure 3. The structure atoms that correspond to the query atoms are rotated/translated onto the pattern points (in this case S4 onto P1, S3 onto P2, and S2 onto P3), and all structure atoms (other than dummies) are checked to see if any are within the radius of any of the outriggers (that is, if any fall into an excluded volume). If none are, the structure is declared a hit. In Figure 4 we see that the would-be hit is rejected because atom S5 "collides" with outrigger R1.

TIME TESTS

We have implemented the search algorithms in a single program 3DSEARCH. A graphics interface allows the user to draw or modify queries, read or write lists of structure identifiers, select databases, and conduct searches.

To illustrate the speed of our system, we will discuss four queries that were derived from pharmacophores found in the literature. (The details of how the queries were derived from the pharmacophores as originally presented in the literature will be published elsewhere.) These queries are shown in Figure 4: H1, the H1-antihistaminic pharmacophore from Borea et al.;[8] CNS, the central nervous system (CNS) pharmacophore of Lloyd and Andrews;[9] ACE, the angiotensin-converting enzyme inhibitor pharmacophore of Mayer et al.;[10] NIC, the nicotinic acetylcholine pharmacophore, which is equivalent to that derived by Beers and Reich,[11] from our earlier work.[2] For this system, we were able to define an agonist-allowed volume and generate outrigger atoms to surround it.[2,12]

These queries are of typical complexity; most pharmacophores can be expressed as five or fewer atoms. One should also note that in some cases, seven or more types may be needed to express the properties of an atom.

The results of searches over the three-dimensional version of our corporate CL File (see previous paper for details) are shown in Table I. The efficiency of the key search is impressive. In the worst case (NIC), in which there was effectively only a single key (since *all* H-bond acceptors in the database have an atom DL attached), the list of possible structures is cut down to 6% of the total database. A key search typically takes less than 1 CPU min on our VAX 8650. The geometric search takes about 0.04–0.10 s per structure. Restricting the search to smaller structures (as in NIC) gives a significant increase in speed. Including outrigger restrictions increases the time, but not unduly so. Although the data are not shown here, we find that including many wild cards in the query slows down the total search time by a factor of 5 or more.

Table I. Results of Searching a Large Three-Dimensional Database (CL File) for Structures That Contain Typical Queries

query[a]	type of search	structures to search	atom count limits[b]	hits	time,[c] s
H1	key	223 988		1268	37
	geom	1 268	none	715	119
CNS	key	223 988		5975	22
	geom	5 975	1–30	163	193
ACE	key	223 988		5431	68
	geom	5 431	none	96	328
NIC	key	223 988		13370	29
	geom	13 370	none	2252	492
	geom	13 370	1–25	1718	276
	geom + outriggers	13 370	1–25	54	340

[a] Query names correspond to those shown in Figure 4. [b] The number of non-hydrogen, non-dummy atoms of a structure must be in this range for it to be considered in the geometric search. [c] VAX 8650.

DISCUSSION

Three-dimensional substructure search systems are not new. One of the first systems, MOLPAT, was discussed by Gund and Wipke[13,14] over 10 years ago. More recently, Willett and co-workers[7,15–17] have published a series of papers describing their own system. They have also reviewed the field.[17] The most recently reported system is ALADDIN, described by Martin et al.[18,19]

Our own system 3DSEARCH is most closely related to that described by Jakes and Willett.[15] Theirs is a three-phase search system: a rapid prescreen based on a direct key scheme, a distance search, and a geometric search using the Ullman algorithm. They have used this system to search subsets (about 12 000 structures) of the Cambridge Structural Database. Important advances of 3DSEARCH over their work are as follows:

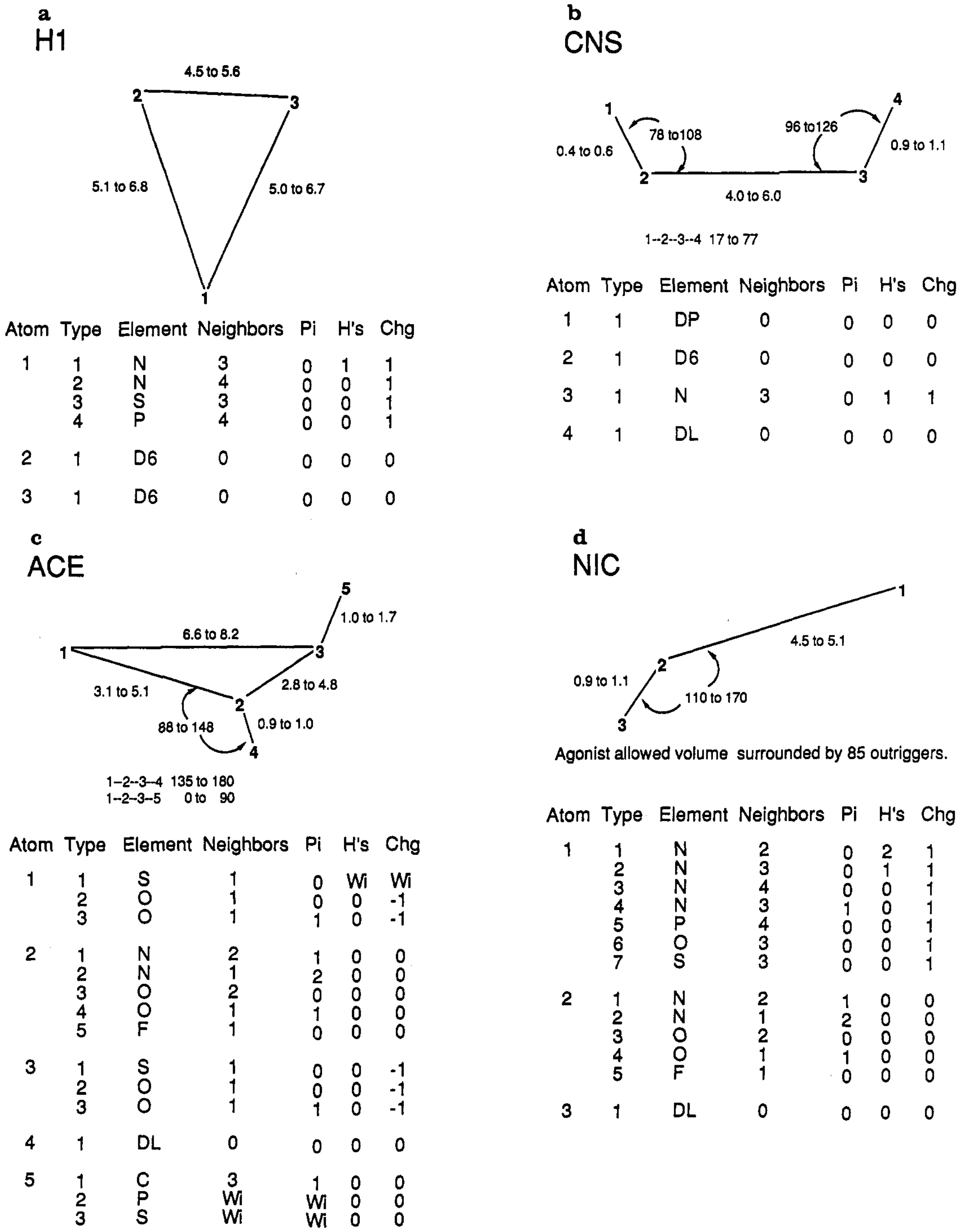
(1) We use a much more specific definition of atom type (five fields instead of two).

(2) We allow multiple types per query atom.

(3) We use a faster inverted key scheme for the prescreen.

(4) We extend the Ullman algorithm to handle angle/dihedral constraints and excluded volume.

It is useful to compare the direct key scheme used by Willett and co-workers to our inverted key scheme, which we believe has important advantages. Like these workers, we take an atom pair (with the atom types and interatomic distance) as the basic substructure with which to define keys. However, the differences in detail as to how the keys are defined and used are important. In a direct key scheme, the number of keys must be fixed at some convenient number and the definition of the keys adjusted so that, for a particular database, the number of structures that contain each key is roughly the same. One consequence of this is that the distance range defined for each key will depend on the frequency of the atom types that appear. (For example, there will be many keys that contain C–C, each with a narrow distance range, while there will be few keys that contain Br–Br, each with a broad distance range.) The consequence of the relatively small number of keys (about 750) used by Willett and co-workers is that the distance ranges in some of the keys are very broad. Thus, a separate distance search is needed to find those structures with a more precise match to the query. In our scheme, since each structure is not associated with a set of keys, the keys need not be limited to a fixed number. We can therefore afford to make the keys very specific. (For instance, we use five fields



Atom	Type	Element	Neighbors	Pi	H's	Chg
1	1	N	3	0	1	1
	2	N	4	0	0	1
	3	S	3	0	0	1
	4	P	4	0	0	1
2	1	D6	0	0	0	0
3	1	D6	0	0	0	0

Atom	Type	Element	Neighbors	Pi	H's	Chg
1	1	DP	0	0	0	0
2	1	D6	0	0	0	0
3	1	N	3	0	1	1
4	1	DL	0	0	0	0

Atom	Type	Element	Neighbors	Pi	H's	Chg
1	1	S	1	0	Wi	Wi
	2	O	1	0	0	-1
	3	O	1	1	0	-1
2	1	N	2	1	0	0
	2	N	1	2	0	0
	3	O	2	0	0	0
	4	O	1	1	0	0
	5	F	1	0	0	0
3	1	S	1	0	0	-1
	2	O	1	0	0	-1
	3	O	1	1	0	-1
4	1	DL	0	0	0	0
5	1	C	3	1	0	0
	2	P	Wi	Wi	0	0
	3	S	Wi	Wi	0	0

Atom	Type	Element	Neighbors	Pi	H's	Chg
1	1	N	2	0	2	1
	2	N	3	0	1	1
	3	N	4	0	0	1
	4	N	3	1	0	1
	5	P	4	0	0	1
	6	O	3	0	0	1
	7	S	3	0	0	1
2	1	N	2	1	0	0
	2	N	1	2	0	0
	3	O	2	0	0	0
	4	O	1	1	0	0
	5	F	1	0	0	0
3	1	DL	0	0	0	0

Figure 4. Four queries derived from pharmacophores in the literature (see text for references). All distances are given in angstroms, and all angles/dihedral angles are given in degrees. Dihedral angles are represented as absolute values. Distances to dummy atoms (elements D5, D6, DP, and DL) reflect the arbitrary distances used to generate the position of dummy atoms when the three-dimensional database was constructed. (a) H1 is the H1 receptor agonist pharmacophore. Atom 1 is a cation; atoms 2 and 3 are centroids for flat 6-membered rings. (b) CNS is the central nervous system pharmacophore. Atom 2 is the centroid of a flat 6-membered ring, atom 3 is a basic tertiary amine, atom 1 indicates the perpendicular of the ring, and atom 4 indicates the direction of the lone pair on the amine. (c) ACE is the angiotensin-converting enzyme inhibitor pharmacophore. Atom 1 is a Zn ligand (sulfhydryl or carboxylate oxygen), atom 2 is an H-bond acceptor, atom 3 is an anion (deprotonated sulfur or oxygen from carboxylate, sulfate, or phosphate), atom 4 indicates the lone-pair direction of atom 2,and atom 5 is the central atom of a carboxylate, sulfate, or phosphate, of which atom 3 is an oxygen, or an unsaturated carbon, where atom 3 is a deprotonated sulfur. (d) NIC is the nicotinic acetylcholine pharmacophore. Atom 1 is a cation, atom 2 is a H-bond acceptor, and atom 3 indicates the lone-pair direction of atom 2. The allowed agonists volume is surrounded by 85 outriggers of 2.5-Å radius.

per atom type and use much narrower distance ranges.) Also, the definition of our keys is independent of the frequency of atom types in the database.

When structures are added to a database, the correspondence of structures with keys must be updated. In a direct key scheme only a few structure records need be added or replaced, while a much more involved process is required for an inverted key scheme. (This advantage of direct keys disappears if a *new* key needs to be added.)

Since searching is a much more frequent operation than updating, the speed advantage of inverted keys is very important, especially for large databases. The time for a direct key search is directly proportional to N, the number of structures in the database. The time for the inverted key search depends on the number of distances specified in the query and on the number of unique keys in the database. This latter number is roughly proportional to log N. This allows us to do a key search over a database of unprecedented size

(>200 000 structures) in less than a minute in most cases.

ALADDIN[18,19] takes a very different approach from ours. It is based entirely on the MEDCHEM software.[20] Connection tables are stored in SMILES[21] linear notation. The corresponding coordinates are stored in a THOR database. Queries are constructed by using the GENIE language, which uses an extension of SMILES to do substructure searches in connection tables. This ability to refer directly to the connection tables gives the user a great deal of flexibility in defining atoms at the query level. Our system, on the other hand, can search only coordinates and atom types; the latter include certain predefined (although probably the most important) types of information extracted from connection tables. ALADDIN can handle distance, angle/dihedral, and excluded volume constraints. An additional use of ALADDIN is in the *generation* of pharmacophore models. The main advantage of 3DSEARCH over ALADDIN is one of speed: ALADDIN searches take several hours; in contrast, 3DSEARCH allows query refinement and searches to be performed interactively.

REFERENCES AND NOTES

(1) Marshall, G. R.; Barry, C. D.; Bosshard, H. E.; Dammkoehler, R. A.; Dunn, D. A. In *Computer-Assisted Drug Design*; Olson, E. C., Christoffersen, R. E., Eds.; ACS Symposium Series 112; American Chemical Society: Washington, DC, 1979; pp 205–226.

(2) Sheridan, R. P.; Nilakantan, R.; Dixon, J. S.; Venkataraghavan, R. The Ensemble Approach to Distance Geometry: Application to the Nicotinic Pharmacophore. *J. Med. Chem.* **1986**, *29*, 899–906.

(3) Crippen, G. M. Distance Geometry Approach to Rationalizing Binding Data. *J. Med. Chem.* **1979**, *22*, 988–997.

(4) Allen, F. H.; Bellard, S.; Brice, M. D.; Cartwright, B. A.; Doubleday, A.; Higgs, H.; Hummelink, T.; Hummelink-Peters, B. G.; Kennard, O.; Motherwell, W. D. S.; Rodgers, J. R.; Watson, D. G. The Cambridge crystal data center: computer-based search, retrieval, analysis, and display of information. *Acta Crystallogr. Sect. B: Struct. Crystallogr. Cryst. Chem.* **1979**, *B35*, 2331–2339.

(5) Rusinko, A., III; Sheridan, R. P.; Nilakantan, R.; Haraki, K. S.; Bauman, N.; Venkataraghavan, R. "Using CONCORD To Construct a Large Database of Three-Dimensional Coordinates from Connection Tables. *J. Chem. Inf. Comput. Sci.* (preceding paper in this issue).

(6) Ullman, J. R. An algorithm for subgraph isomorphism. *J. Assoc. Comput. Mach.* **1976**, *23*, 31–42.

(7) Brint, A. T.; Willett, P. Pharmacophore pattern matching in files of 3D chemical structures: comparison of geometric searching algorithms. *J. Mol. Graphics* **1987**, *5*, 49–56.

(8) Borea, P. A.; Bertolasi, V.; Gilli, G. Crystallographic and conformational studies on histamine H1-receptor agonists. *Arzneim.-Forsch./Drug Res.* **1986**, *36*, 895–899.

(9) Lloyd, E. J.; Andrews, P. R. A Common Structural Model for Central Nervous System Drugs and Their Receptors. *J. Med. Chem.* **1986**, *29*, 453–462.

(10) Mayer, D.; Naylor, C. B.; Motoc, I.; Marshall, G. R. A unique geometry of the active site of angiotensin-converting enzyme consistent with structure–activity studies. *J. Comput.-Aided Mol. Des.* **1987**, *1*, 3–16.

(11) Beers, W. H.; Reich, E. Structure and activity of acetylcholine. *Nature* **1970**, *228*, 917–922.

(12) Sheridan, R. P.; Venkataraghavan, R. Designing novel nicotinic agonists by searching a database of molecular shapes. *J. Comput.-Aided Mol. Des.* **1987**, *1*, 243–256.

(13) Gund, P.; Wipke, W. T.; Langridge, R. Computer searching of a molecular structure file for pharmacophoric patterns. In *Proceedings of the International Conference on Computers in Chemical Research and Education*, Ljubljana; Elsevier: Amsterdam, 1974; Vol. 3, pp 5/33–38.

(14) Gund, P. Three-dimensional pharmacophore pattern searching. *Prog. Mol. Subcell. Biol.* **1977**, *5*, 117–143.

(15) Jakes, S. E.; Willett, P. Pharmacophoric pattern matching in files of 3-D chemical structures: selection of interatomic distance screens. *J. Mol. Graphics* **1986**, *4*, 12–20.

(16) Jakes, S. E.; Watts, N.; Willett, P.; Bawden, D.; Fisher, J. D. Pharmacophoric pattern matching in files of 3D chemical structures: evaluation of search performance. *J. Mol. Graphics* **1987**, *5*, 41–48.

(17) Brint, A. T.; Mitchell, E.; Willett, P. Substructure searching in files of three-dimensional chemical structures. In *Chemical Structures: the International Language of Chemistry*; Warr, W. A. Ed.; Springer-Verlag, Berlin, 1988; pp 131–144.

(18) Martin, Y.; Danaher, E. B.; May, C. S.; Weininger, D. MENTHOR, a database system for the storage and retrieval of three-dimensional molecular structures and associated data searchable by substructural, biologic, physical, or geometric properties. *J. Comput.-Aided Mol. Des.* **1988**, *2*, 15–29.

(19) Van Drie, J. H.; Weininger, D.; Martin, Y. C. ALADDIN: An integrated tool for computer-assisted molecular design and pharmacophore recognition from geometric, steric, and substructure searching of three-dimensional molecular structure. *J. Comput.-Aided Mol. Des.* **1989** (in press).

(20) Weininger, D.; Weininger, A.; Leo, A. J. *MedChem Software Manual*, Release 3.52; Medicinal Chemistry Project, Pomona College, Claremont, CA, 1987.

(21) Weininger, D. SMILES, A Chemical Language and Information System. 1. Introduction to Methodology and Encoding Rules. *J. Chem. Inf. Comput. Sci.* **1988**, *28*, 31–36.