Structure Searching in Chemical Databases by Direct Lookup Methods†

Bradley D. Christie, Burton A. Leland, and James G. Nourse*

Molecular Design Limited, 2132 Farallon Drive, San Leandro, California 94577

Received October 1, 1992

Modern chemical structure databases represent a continuing challenge to efficient and accurate searchability. As these databases grow in size and complexity, it is important to develop search methods that are less dependent on database size. Direct lookup methods can have the effect of being largely independent of database size. Over the years at Molecular Design Ltd. we have developed methods to do exact structure search, "near exact" structure search, and substructure search which approach the ideal of direct lookups.

INTRODUCTION

As the use of computer databases for storing chemical structure data has become the method of choice, new challenges for search and retrieval arise from both the size of these databases as well as the complexity of the data. Both public and private databases can now contain millions of separate entries. These entries may be simple chemical structures (Figure 1) or complex chemical substances (Figure 2). Any entry may have an arbitrary amount of data associated with it. In the case of chemical substances, there may be data associated with individual parts of the various structures.[1] Accurate and efficient searching of such databases represents a continuing challenge. This challenge has been addressed by a number of workers over the years at Molecular Design Ltd.[2] This paper will summarize some of this work.

EXACT STRUCTURE MATCH

Once a chemical structure has been registered into a database, the question of retrieval of this structure using a query which contains exactly the same structure immediately arises. We have called this a "find current" search since at the user level the database is queried using the structure currently visible to the user. This search has been traditionally done by representing the structure with a unique name or bitstring.[3] The query structure is given the same name, and through a hash, retrieval, and bitstring comparison, the desired structure can be retrieved. This is shown symbolically in Figure 3. The long horizontal bar represents the entire database with the few desired structures represented by vertical lines. The heavy arrow represents a direct lookup to yield a much smaller set of candidates. In this case this direct lookup is done by a hashing method. The arrows at each end of this smaller horizontal bar represent a procedure which must deal with each candidate individually in turn to yield the final set of exact match structures. In this case the final procedure is a bitstring comparison. Because a direct lookup is possible, the search proceeds effectively, efficiently, and nearly independently of the size of the database. Only if the database contained mostly examples of the desired structure of if extensive hash collisions were seen would a dependence on database size be noticeable.

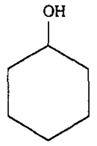

Figure 1. Chemical structure.

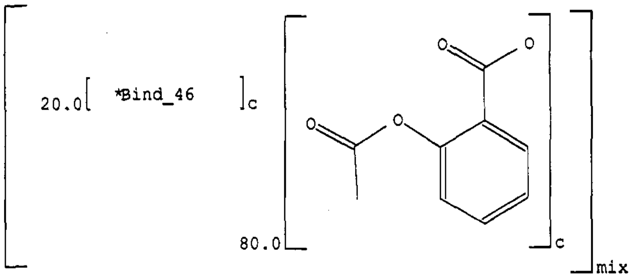

Figure 2. Chemical substance. This represents a mixture of aspirin and an unstructured binder in a ratio of 80:20.

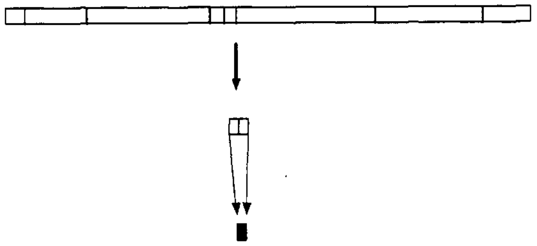

Figure 3. Exact structure search. This search proceeds by a hash lookup followed by individual candidate verification. See text for explanation of symbols.

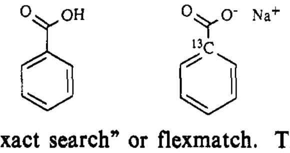

Figure 4. "less exact search" or flexmatch. This is done to find nearly equivalent structures such as salts and neutral species.

While an exact structure search is necessary, the existence of the capability to find only structures which exactly match the query structure in every detail is not sufficient for all anticipated needs of this type. For example the query may be an uncharged species and the database structure a salt. These are certainly not exactly the same in the usual sense, yet the need for one to find the other is often critical. Other variations between query and structure could include isotopic substitution, stereochemistry, tautomerism, etc. We have addressed this need by introducing a feature called flexmatch (Figure 4). The flexmatch search also proceeds by a hashing method except that, instead of a complete structure name, a set of structural features for individual fragments are perceived and hashed (Figure 5). The set of candidate structures retrieved this way is then compared to the query structure using an atom-by-atom verification (Figure 6). Only the parts

† Presented in a symposium on Structure Searching given by the ACS division of Computers in Chemistry at the 204th National Meeting of the ACS in Washington, D.C., on Aug 24, 1992.

0095-2338/93/1633-0545$04.00/0

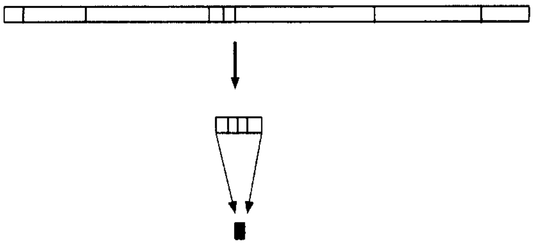

Figure 5. Flexmatch search. This search proceeds by a hash lookup followed by individual candidate verification.

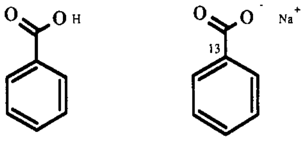

Figure 6. Flexmatch candidate verification, is done by an atom-by-atom match. The common substructure verified is emphasized.

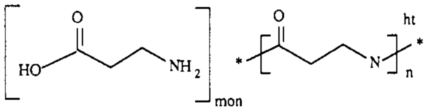

Figure 7. "Even less exact search", done to find polymer monomers and structural repeating units.

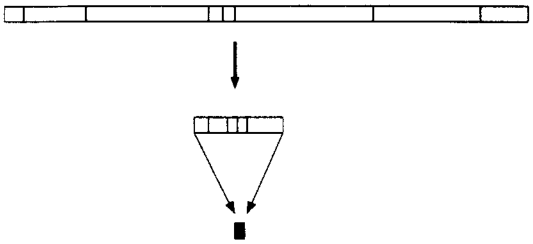

Figure 8. Monomer-SRU search. This search proceeds by a several step hash lookup followed by individual candidate verification.

of the structure common to both the query and database structure are found in the atom-by-atom verification. This is indicated by the darkened parts of the structures in Figure 6. At the user level there are presently 18 choices for structural features which can be considered or ignored in the flexmatch search.[1] Since a hash lookup is used the dependence on database size in theory is minimized. In practice a larger number of candidates must be verified in flexmatch than in exact match search. This extra verification is necessary because there are more correct candidates, candidates which are not distinguished by the features hashed, and hash collisions.

An additional need for a "near exact" search arises by the need to search polymeric structures. Such structures can be represented as source-based monomers or polymer-based structural repeating units (SRU) (Figure 7). While the difference between the two is a chemical reaction, the nature of the reaction is often not an issue or completely unknown and only static structural data are involved. This type of searching is also done by a hash lookup and atom-by-atom verification (Figure 8). In this case the hash is done on even fewer structural properties since not all atoms or bonds will be present in both the monomer and SRU form. The atom-by-atom verification will find only parts of the structures which are common (Figure 9). This type of searching works for addition polymers as well as some common condensation polymers.[1]

These modifications of the "exact structure" search allow us to continue to take advantage of the very fast direct lookup methodology which is effectively independent of database size. However, less and less of the structure is used to create an effective hash which in some worst cases could lead to very large numbers of candidates verified by atom-by-atom matching. This approach is of limited use for more general partial structure queries.

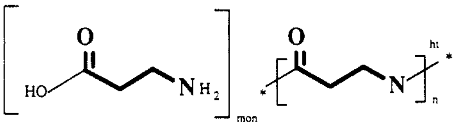

Figure 9. Monomer-SRU verification, done by an atom-by-atom match. The common substructure is emphasized.

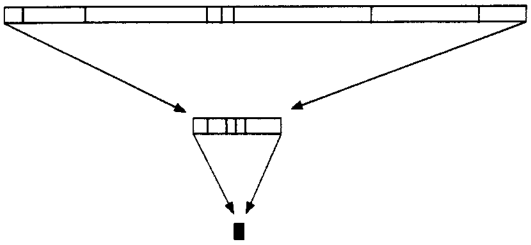

Figure 10. Our first substructure search system. This required two sequential scans of large numbers of structures.

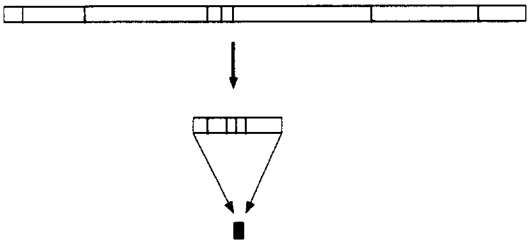

Figure 11. Our current substructure search system. It uses one direct lookup and one sequential scan of the remaining candidates.

SUBSTRUCTURE SEARCH

Substructure search is the traditional method for querying chemical structure databases with only partial structures as queries. While this problem is fundamentally different from the exact structure search problem, it is possible to look at various approaches in terms of the number of candidates found by direct lookup versus individual structure verification. Over the years our approach to this problem has evolved as the sizes of databases and user needs have grown.

Our first substructure search system used a traditional method of fragment based screens or keys. A database entry and a substructure query had the same keys set. This set of keys for the query was reduced to a bitstring and then compared with the bitstring retrieved for every structure in the database (Figure 10). Database structures with every one of the query keys set were saved as candidates. The candidates which survived were retrieved and verified by an atom-by-atom search. Thus in this system large numbers of structures were dealt with individually two times, as shown in Figure 10.

Our subsequent and present system uses a much more comprehensive set of keys which are inverted as bitstrings and stored. Each key becomes one bitstring whose length is the size of the database. The inverted key bitstring is retrieved for each key set by a query. These are combined by a logical AND operation to yield the candidate structures (Figure 11). This provides a direct lookup method for retrieving the candidates, although there is a database size dependency since the bitstrings for each key are the size of the database. These candidates are each retrieved individually and verified by atom-by-atom search. All candidates must be verified, and even if

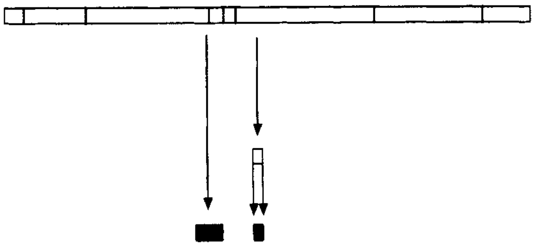

Figure 12. Our developing substructure search system. It finds nearly all desired structures by direct lookups.

the screenout is good, this still can be a very time consuming operation when there is a large number of correct structures. In this system large numbers of structures are dealt with individually only one time since the screenout is by a direct lookup, as shown in Figure 11.

We are currently working on a new system to further improve searching performance. The approach is derived from a research system devised earlier for carbon-13 NMR spectra assignment.[4] A unique atom-centered name is created for each atom of each structure in the database. These are stored in a complex index file. Equivalent partial environments are stored only once. A query has one or more nonunique atom centered names created, and, with extensive consideration for the fact that queries are not complete structures, the query structure is "looked up" in the index file. A single lookup of this type generally leads to most if not all of the database structures with the desired substructure present. Occasionally candidates are retrieved which require an atom-by-atom verification (Figure 12). With the exception of these few structures, this system proceeds without ever dealing with large numbers of structures individually, as shown in Figure 12.

CONCLUSION

Direct lookup methods provide a means of searching for chemical structures largely independently of database size. This progression of methods using traditional hashing and more elaborate lookup methods for exact, near exact, and substructure search has contributed to solutions to the problems associated with searching large, complex chemical structure databases.

REFERENCES AND NOTES

(1) Gushurst, A. J.; Nourse, J. G.; Hounshell, W. D.; Leland, B. A.; Raich, D. G. The Substance Module: The Representation, Storage, and Searching of Complex Structures. *J. Chem. Inf. Comput. Sci.* **1991**, *31*, 447–454.

(2) Besides the authors these include: J. Barstow, R. Briggs, J. D. Dill, D. L. Grier, A. J. Gushurst, A. K. I. Gushurst, D. Henry, W. D. Hounshell, J. Laufer, T. E. Moock, D. G. Raich, and W. T. Wipke.

(3) Wipke, W. T.; Dyott, T. M. Stereochemically Unique Naming Algorithm. *J. Am. Chem. Soc.* **1974**, *96*, 4834–4842.

(4) Gray, N. A. B.; Nourse, J. G.; Crandell, C. W.; Smith, D. H.; Djerassi, C. Stereochemical Substructure Codes for 13C Spectral Analysis. *Org. Magn. Reson.* **1981**, *15*, 375–389.