Protein Engineering vol.7 no.1 pp.23–29, 1994

A novel search method for protein sequence–structure relations using property profiles

Gerrit Vriend, Chris Sander and Pieter F.W.Stouten[1]

European Molecular Biology Laboratory, Meyerhofstrasse 1, D-6900 Heidelberg, Germany

[1]Present address: Du Pont Merck Pharmaceutical Company, PO Box 80353, Wilmington, DE 19880-0353, USA

In protein engineering and design it is very important that residues can be inspected in their specific environment. A standard relational database system cannot serve this purpose adequately because it cannot handle relations between individual residues. With SCAN3D we introduce a new database system for integrated sequence and structure analysis of proteins. It uses the relational paradigm wherever possible. Its main power, however, stems from the ability to retrieve stretches of consecutive residues with certain properties by comparing a property profile with all stretches of residues in the database, exploiting the ordered character of proteins. In doing so, it bypasses the large number of join operations that would be required by relational database systems. An additional advantage of using property profile matching is that searches can be carried out allowing a pre-set number of mismatches. Also, as the database is read-only, SCAN3D does not need interactive data update mechanisms. Queries typical of a molecular engineering environment are demonstrated with specific examples: analysis of peptides that induce local structure, analysis of site-dependent rotamers and residue–residue contact analysis.

Key words: fast retrieval/property profile matching/protein database/relational approach/sequence–structure analysis

Introduction

In the field of protein engineering, careful analysis of protein sequences and their 3-D structures can lead to an improved understanding of these proteins. However, the elucidation of a protein structure by X-ray or NMR methods often generates as many questions as it answers. In such cases multiple protein structure comparisons provide a powerful tool for revealing and understanding general and specific features of protein structures. Examples of such comparative studies are abundant in the literature (e.g. McGregor *et al.*, 1987; Ponder and Richards, 1987; Richardson and Richardson, 1988; Baumann *et al.*, 1989; Vriend and Sander, 1991). Analysis of helix caps (Richardson and Richardson, 1988) has produced rules that can be used to predict thermostabilizing mutants in proteins; analysis of protein surfaces (Baumann *et al.*, 1989) has led to an algorithm to evaluate the 'normality' of (designed) proteins and study of side chain torsion angles (McGregor *et al.*, 1987; Ponder and Richards, 1987) has provided insight into the packing of protein cores and has played a role in the design of a computer program that can rebuild protein structures from Cα coordinates only (Holm and Sander, 1991). These comparisons help us to establish the relationship between protein sequence and structure and to reveal common patterns and features in both sequences and structures.

It is obvious that database systems that allow for fast, easy and flexible retrieval of specific information are crucial for studying the principles that determine protein structure. Several general (Bryant, 1989; Islam and Sternberg, 1989; Gray *et al.*, 1990; Vriend, 1990a; Huysmans *et al.*, 1991) and single-purpose (Lesk *et al.*, 1989; Sander and Schneider, 1991) data storage and retrieval systems have been developed to analyse protein sequences and structures. Some of them hardly (re)organize data, but merely combine a database of 3-D protein structures with a set of algorithms for pattern recognition, data analysis and graphics. In general, these systems provide very flexible tools, but this flexibility is paid for by rather low speed when the algorithms are applied to large amounts of data. PKB (Bryant, 1989) and to a certain extent the parameter correlation method (Vriend, 1990a) are good examples of such systems. They are well suited for prototyping queries or searches in small subsets of the database, but less suitable for use in an environment where short query turn-around times are required.

If retrieval times must be reduced to a minimum, one resorts to systems that pre-process and reorganize data in order to speed up the process of extracting information. Two important classes of such systems are object-oriented database systems (OODBS) and relational database systems (RDBS) (Parsaye *et al.*, 1989). An OODBS can easily search for many related objects, but the organization of the data makes it slow at doing sequential scans (Gray *et al.*, 1990). P/FDM (Gray *et al.*, 1990) is a good example of an OODBS. Its high level query language Daplex is very concise and approaches the power of a programming language for complex queries.

In a protein RDBS many structural properties such as accessibility, torsion angles and secondary structure are stored in tables and queries are performed by logical combination of these tables. BIPED (Islam and Sternberg, 1989) and SESAM (Huysmans *et al.*, 1991) are examples of such systems. SESAM does not fit the relational model exactly as it also provides some algorithms on top of the RDBS to allow for otherwise impossible or prohibitively slow queries. Advantages of a generalized RDBS (Parsaye *et al.*, 1989) are the generally high speed of searches and the intuitive way in which queries are constructed.

When one wants to use a standard RDBS for protein analysis, three major problems are encountered.

(i) The database management program generates much unnecessary overhead. A standard RDBS assumes that its data are unsorted and it is generally not possible to bypass its sorting and data (re)organization routines. However, proteins are well organized and have a fixed order: the N-terminal residue always comes first and the C-terminal residue last. A general RDBS has provisions for adding, deleting and modifying data at run-time, which requires internal safety mechanisms. In general, however, modification of a database with protein 3-D structures is necessary only a few times a year, e.g. when the new PDB (Bernstein *et al.*, 1977) is released.

(ii) Entries in the same property column are assumed to be unrelated but for protein structures this is not the case. In fact,

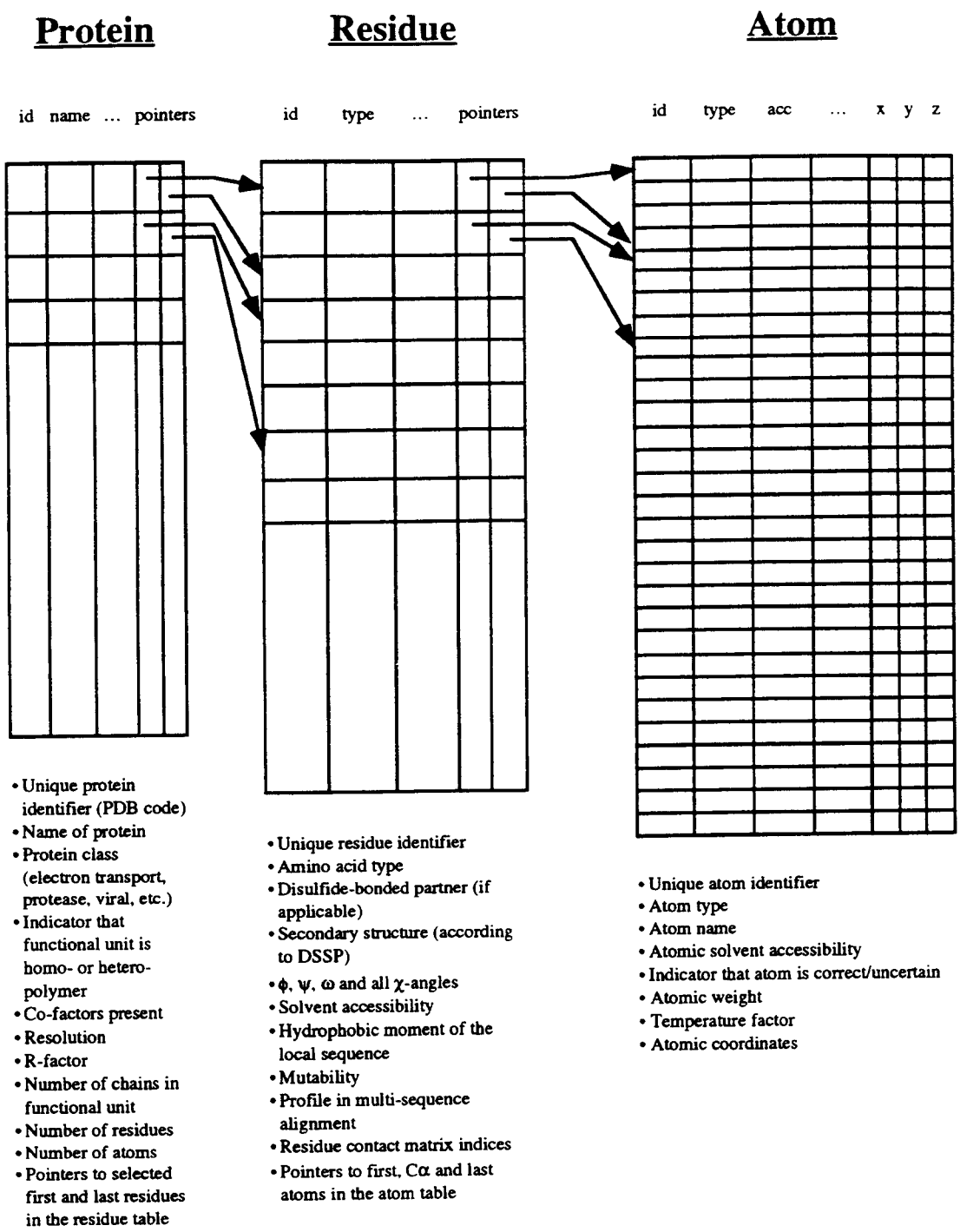

Fig. 1. The hierarchical, three-level organization of the SCAN3D database tables. The properties stored in their columns are listed underneath. 'id' represents identifier and 'acc' represents accessibility. The 'pointers' establish the links from the highest level (protein) via the residue level down to the lowest level (atom). Contacts between amino acids are very important for protein structure analysis. They are not single-residue properties and are difficult to represent in a standard relational database. In SCAN3D they are handled using one column of pointers in a contact network ('residue contact matrix indices' in the residue table).

the most relevant queries concern relations between residues separated by a given distance in sequence and these are inherently beyond the capabilities of a standard relational system (Islam and Sternberg, 1989).

(iii) New types of searches that were not foreseen during the design of the database are generally not possible to do in an RDBS.

SCAN3D was specifically designed to solve or bypass these problems. Consequently, we did not adhere strictly to the paradigm of relational databases. SCAN3D uses the power and speed of such approaches wherever possible, but all unnecessary overhead has been suppressed. SCAN3D exploits the sorted character of protein structures and the fact that there is no need for the user to modify or add data. The problem of relations between entries in the same column is bypassed by matching search profiles (stretches of consecutive residues with certain characteristics) to the database tables rather than doing individual table look ups (see below).

Design, implementation and performance

Data organization

The Brookhaven protein databank (PDB) (Bernstein *et al.*, 1977) contains atomic coordinates and some related information for more than 1000 macromolecular structures in plain text files. SCAN3D uses a limited, representative set of slightly more than 200 protein structures (Hobohm *et al.*, 1992). This is done to avoid bias towards the small number of abundantly present protein families. For example, the June 1992 version of the PDB contains more than 100 lysozyme, 30 haemoglobin and 10 rhinovirus (with three related chains each) structures. The PDB files with crystal structures often contain errors, such as residues with wrong chirality (Morris *et al.*, 1992), the absence of atoms and residues, atoms with incorrect names, histidines with six membered rings, etc. The structure input modules of the WHAT IF program (Vriend, 1990b) adequately deal with these problems by issuing warnings and optionally applying corrections. Hereafter, accessibility for solvent, secondary structure, mutability, torsion angles, position in the chain and other properties are derived from PDB files in an automated fashion and stored in the database tables. The hierarchical, three-level organization of the SCAN3D data tables is described in Figure 1.

Profile matching

Many queries that concern single residue properties can be handled efficiently by a general protein RDBS. Queries such as 'find all prolines with a positive ϕ-angle' require two-table look-ups and an AND operation and responses to such queries are generally very fast. However, the analysis of protein structures and sequences most often leads to queries that concern characteristics of neighbouring residues or residues that are a certain sequence distance apart. Examples are 'find all prolines that do not have another proline within 27 residues in the sequence' or 'find all prolines that have an atomic contact between their side chain and a cysteine Sγ which is part of a disulphide bond'. Such queries are much more difficult or sometimes even impossible to perform when using a strictly relational system. For example, in order to obtain information about the sequence spanning 15 residues on either side of the one being inspected, one would need 30 additional columns [as in BIPED (Islam and Sternberg, 1989)] or 30 join operations to the residue table to pick up these values [as in SESAM (Huysmans *et al.*, 1991)]. Another 30 join operations would be needed to retrieve the solvent accessibility of those residues, etc. Clearly this is not the most efficient approach.

SCAN3D solves this key problem of relational approaches by comparing a profile of the desired length and with the desired residue properties with all stretches of that length in the entire database. The answer to a query is a group of hits (GOH), i.e. a list of residue stretches that satisfy the constraints imposed by the profile on the individual residues. SCAN3Ds GOHs can be combined with logical operators such as AND, OR and NOT. AND should read as 'is present in both GOHs' and OR as 'is present in at least one GOH'. Internally these operations are performed using a standard sort–merge-like algorithm. Because the data in each GOH are always sorted this algorithm works extremely fast. This feature of comparing more than one residue at a time gives SCAN3D a great deal of its power.

Figure 2 shows an example of a search for sequences that satisfy certain constraints. The stretches to be found should have the following sequence profile, G-aromatic-any-G-small-G, where 'any' stands for any of the 20 residues and G for glycine. The leftmost columns in this figure represent the GOH generated by

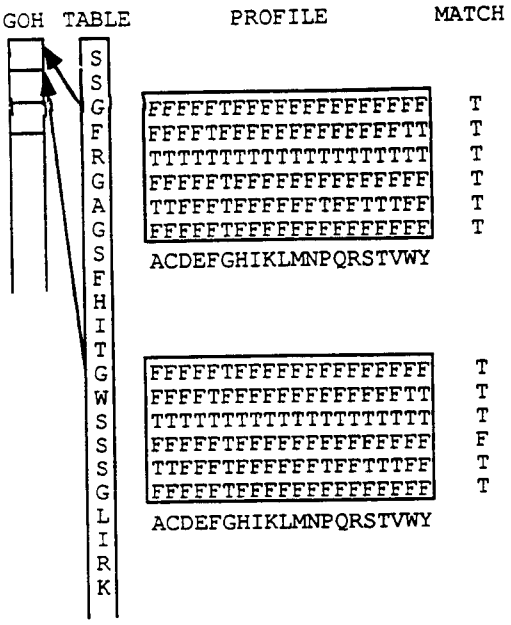

Fig. 2. Example of a sequence profile search. Two of the matches between the data table and the sequence profile are shown. The second profile match would only enter the group of hits (GOH) if the allowed number of mismatches is set to 1 or more. The amino acid string at the bottom is given for easy reference only.

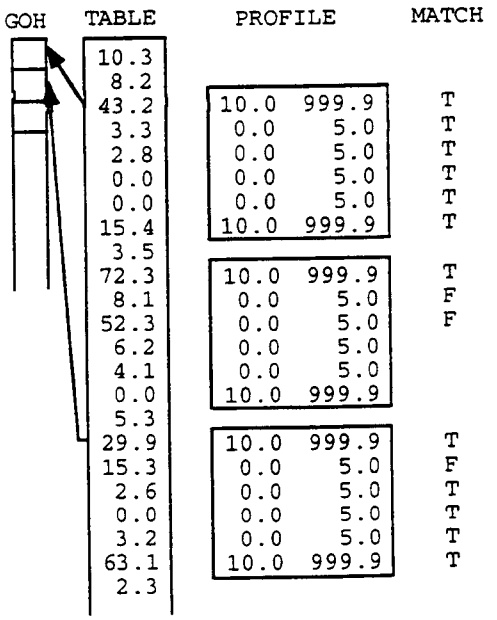

Fig. 3. Example of a search for four buried residues flanked on either side by an exposed residue. Three of the matches between the data table and the accessibility profile are shown. In this example the profile boxes contain lower and upper bounds of accessibility rather than Boolean values. Assuming that the allowed number of mismatches had been set to 1, only the first and third examples will enter the group of hits (GOH).

this query and the protein sequence table that is being inspected. The stretch of amino acids searched for is six amino acids long and, thus, the profile box is dimensioned 6 × 20. Assuming that the profile width is less than the length of the shortest sequence in the table, $N \times (M-P+1)$ profile matches have to be performed, where N is the number of proteins in the database, M the mean length of each protein and P the profile width (6

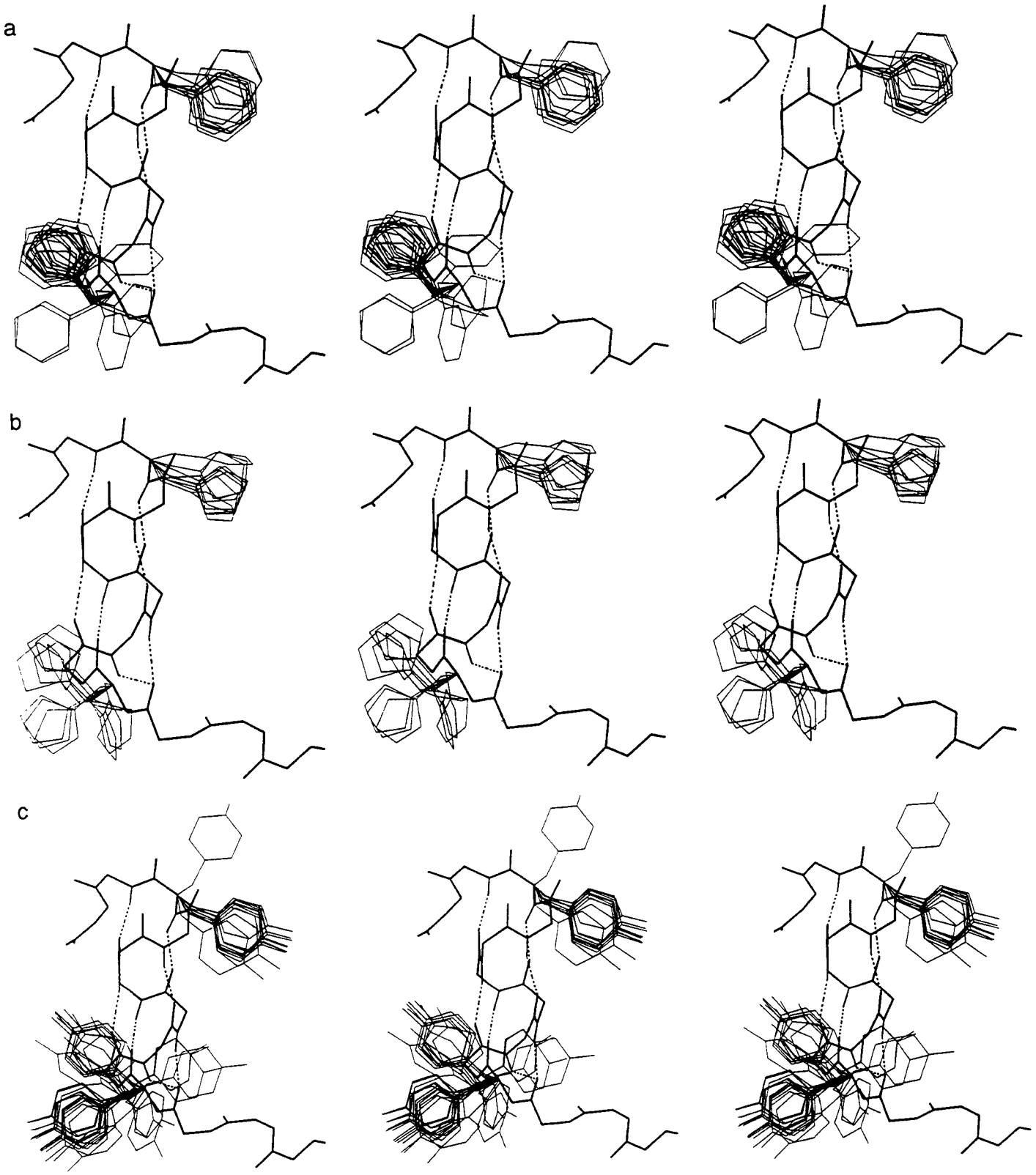

Fig. 4. χ_1 rotamer distributions for Phe (a), His (b) and Tyr (c) in helices. Note the differences in the position-specific rotamer distributions at the ends of helices.

in this example). Two of the hexapeptide stretches that meet the constraints are indicated in Figure 2. The top profile box matches the corresponding amino acids in the sequence table perfectly and, consequently, a pointer to this stretch of amino acids is stored in the GOH. The second profile box shows one mismatch (the glycine at position 4 in the profile is a serine in the sequence table). A pointer to this stretch of amino acids would therefore only be stored in the GOH if the user-defined number of allowed mismatches is 1 or more.

The same principle can be used for secondary structure

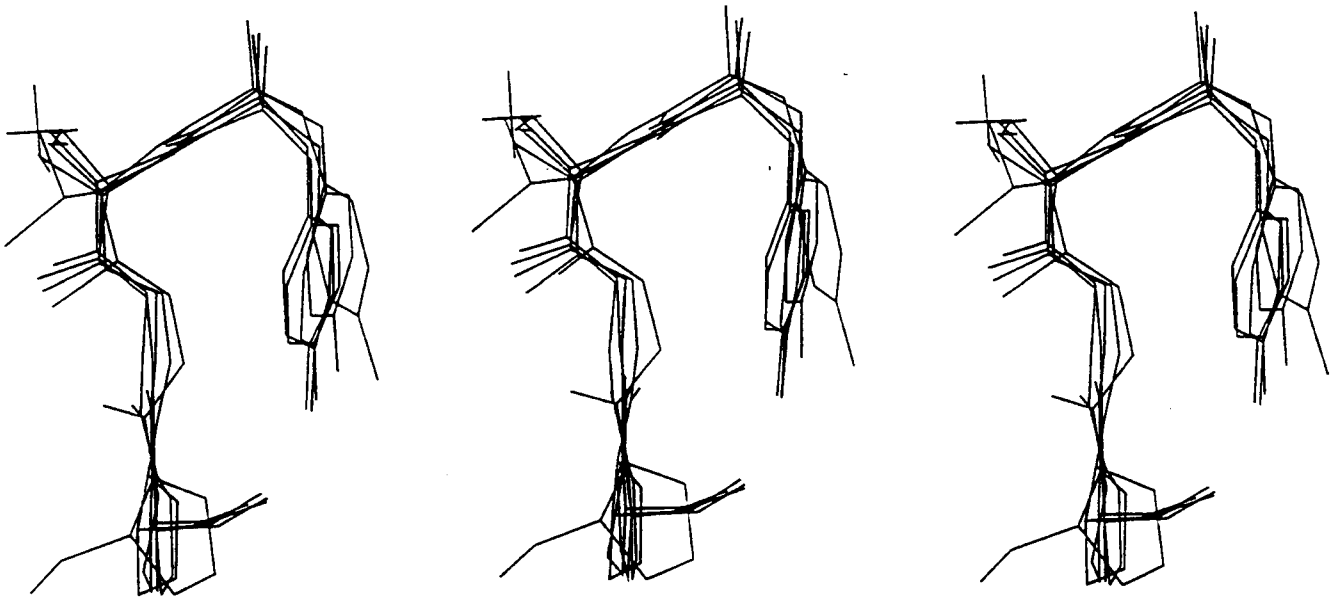

Fig. 5. Conformations of Y(S/T)GP tetrapeptides. Superimposed are the conformations of all five tetrapeptides of sequence Y(S/T)GP. They were found in 1CHB (residues 13–16), ITGP (10–13), 3RHV (117–120), 4XIA (284–287) and 6XIA (279–282). The example illustrates that the intrafragment contacts found in small peptides are also seen in native proteins.

searches. Here the table stores a code for each residue that represents helix, strand, turn or coil. The profile no longer contains 6 × 20 true/false flags (a six-residue stretch with 20 possible amino acids at each position), but only 6 × 4 (a six-residue stretch with four possible states for each residue).

A slightly different principle is used for numerical tables. Figure 3 shows how a search for four buried residues (accessible surface <5 Å^2) flanked at both ends by an exposed residue (accessible surface >10 Å^2) is performed. It is not possible to match this profile in exactly the same fashion as in the previous example because accessibility is more adequately represented by a real number than by a logical value. A match is found when the accessibility of the residue falls within the limits set by the profile. In Figure 3, three of the $N \times (M - P + 1)$ possible profile box positions are indicated. The top profile box shows an exact match. The bottom box shows a match with just one mismatch. The middle box shows an example where the program does not have to evaluate the entire profile because the actual number of mismatches (two in this box) exceeds the allowed number of mismatches (one in this example).

Local folds are important for the analysis of proteins and, therefore, SCAN3D allows the user to search for stretches of amino acids that have a desired backbone and/or side chain conformation. The fragment databases required to perform such searches have been described by Jones and Thirup (1986) and Levitt (1992). Results of these searches can be combined with the results of other queries and this opens a new range of possibilities (as shown in example 1 below).

Resource demands and performance

The present version of the SCAN3D database contains 242 structures within a total of 48 906 residues. The atomic data occupy 10 MB of disk space. Another 15 MB is occupied by all pointer and distance tables for the structure fragments. The atomic contact network information takes 75 MB. All other tables together occupy approximately 3 MB. Most simple queries are performed in a matter of seconds or less on a small workstation. For example, the search depicted in Figure 2 took 0.4 s c.p.u. on a Silicon Graphics Indigo workstation including all overheads. The c.p.u. time for logical operations on GOHs is ~0.1 s.

Searches for 3-D contacts, such as those described by Singh and Thornton (1990), are among the most complex operations from the program's point of view. Searching all prolines with their Cδ in contact with a tyrosine Cγ, which is an example of such a complex operation, takes SCAN3D ~30 s c.p.u. on a small workstation.

Examples

It is impossible to describe all possible queries that can be performed by combination of the search options provided by SCAN3D. Rather, several typical queries will be described. The first example elucidates position-dependent rotamer preferences. This query is routinely executed when designing point muations or when building a model by homology. The second example describes use of the contact database. The nature and frequency of contacts between lysine Nζ and aromatic rings are revealed and compared with literature data (Burley and Petsko, 1986; Levitt and Perutz, 1988). The third example reveals that very short residue stretches can induce a unique local fold. The final (fourth) example shows how relevant conclusions about the structure of a retention signal fragment can be drawn on the basis of its sequence.

Analysis of site-dependent rotamer preferences

Many efforts at protein structure prediction have focused on side chain conformations as a tangible subproblem (McGregor *et al.*, 1987; Ponder and Richards, 1987; Holm and Sander, 1991). The ability to accurately predict side chain conformations (rotamers), given the sequence and main chain fold, is very useful for homology modelling (McGregor *et al.*, 1987; Levitt, 1992). McGregor *et al.* (1987) evaluated the χ_1 torsion angles as a function of secondary structure. They observed clear differences in the population of the three preferred χ_1 areas when comparing residues in different types of secondary structure. However, when using secondary structure-based χ_1 distributions in model-building procedures two problems are encountered.

First of all, the terminal residues of secondary structure

elements are often not uniquely defined and different secondary structure classification methods place them at different residues. A good example is described by Eijsink *et al.* (1992), where we studied the contribution of helix caps to the thermal stability of neutral proteases. In the majority of cases the three different methods used [our own based on local hydrogen bonding patterns, the DSSP program (Kabsch and Sander, 1983) and the method described by Richardson and Richardson (1988)] disagreed about the positions of helix ends. So, depending on the actual method to determine the ends of helices, χ_1 angles for residues near the termini of helices would be taken from different rotamer distributions. Second, it is too much of a simplification to regard χ_1 distributions as secondary structure dependent.

Here, these problems are elegantly circumvented by searching for rotamers in the context of a specific backbone trace of typically seven residues length. Figure 4 shows predicted rotamers for two positions in the second helix (residues 23–30) of crambin (Hendrickson and Teeter, 1981). The rotamer distributions in this figure were determined by extracting from the database all fragments of five residues that fit to two selected five-residue stretches in crambin with a Cα r.m.s. error of 0.5 Å or less. In addition, the central residue was constrained to be phenylalanine, histidine or tyrosine (see Figure 4). The χ_1 distributions are clearly position specific (they differ significantly for the two positions in the helix) and residue specific (the three residues have different distributions, especially at the N-terminal end). It is beyond the scope of this paper to explain the differences between these six distributions. Briefly, helix capping and the preference for hydrophobic residues to be buried, combined with the fact that the ends of helices are often at the surfaces of proteins, are probably important factors. Steric hindrance between the main chain and the side chain most likely dictates the limited range of allowed conformations for all three residues at the C-terminal end of the helix.

We conclude that for initial placement of side chains in model-building procedures site-specific rotamer usage using fragment retrieval has a clear advantage over secondary structure-dependent methods.

Analysis of very short residue stretches that induce local structure

NMR studies of short Y-T-G-P-containing peptides (residues 11–14 of the small protein bovine pancreatic trypsin inhibitor, BPTI) consistently revealed a short distance (NOE constraint) between the tyrosine ring system and the glycine NH (Kemmink *et al.*, 1993). In order to investigate whether the conformation implied is a general feature of tetrapeptide stretches that resemble Y-T-G-P, we retrieved tetrapeptides such as F-small-G-P, Y-S/T-G-small, F-S/T-G-P and Y-S/T-G-G and analysed their structures. In general, no unique conformation was found. However, the search for Y-S/T-G-P gave five hits that were all in the Y-T-G-P NMR conformation (Figure 5). These Y-S/T-G-P fragments were found in proteins as different as rhinovirus, xylose isomerase, BPTI and cellobiose hydrolase. The corresponding F-S/T-G-P fragment search did not produce a unique conformation, indicating that both the interaction between G13 NH and the ring of Y11 (see also next example) and the hydrogen bond-forming abilities of Y11 OH stabilize the special Y-T-G-P NMR conformation. The surprising observation that only four residues can induce a unique local fold suggests that nucleation of protein folding may involve very short residue stretches.

Analysis of contacts between lysine Nζ and aromatic rings

The previous example raises the question whether close contacts between amino groups and phenyl rings are a general feature in protein structures. Hydrogen bonds between amino groups and phenyl rings have been predicted (Levitt and Perutz, 1988) and observed more frequently than could be expected from a random distribution (Burley and Petsko, 1986) . This is supposed to be due to a favourable electrostatic interaction of the partial positive charge on the hydrogen atoms of the amino groups and the π-electron cloud of the rings. In order to investigate to what extent these polar interactions are important we searched such close contacts taking lysine as a test case. Figure 6 shows all lysine side chains whose Nζ atom is within contact distance (5.0 Å) of the centre of any atom of a phenyl ring. Although in total more than 70 hits are found, no more than two of those have the lysine Nζ close to the centre of the ring. Figure 6 shows that most often the lysine side chains are oriented roughly parallel to the plane of the ring. This, in fact, suggests that it may be energetically more favourable for the ring to engage in dispersion interactions with the aliphatic part of the lysine side chain than in electrostatic interactions with the charged end. Combined with the fact that the aromatic–Nζ interaction energy is much smaller than the hydrogen bonding energy of the Nζ with, for example,

Fig. 6. Contacts between lysine Nζ and aromatic rings. Superimposed are all aromatic rings and lysine side chains, whose Nζ atoms are in contact with these aromatic rings. An Nζ atom is in contact with an aromatic ring if it is <5 Å away from the centre or any atom of this ring. Note the directional anisotrophy of the distribution.

water, we conclude that the hydrogen bonds between lysine Nζ and phenyl rings are not likely to constitute an important factor in protein structures.

Characterization of a retention signal structure

The experiments of Munro and Pelham (1987) have greatly improved our understanding of the retention mechanism of soluble resident proteins in the endoplasmic reticulum (ER). They showed that the addition of six residues (S-E-K-D-E-L) to lysozyme mediated its retention in the ER. The sequences of a number of other retained ER proteins as well as mutagenesis studies have confirmed the validity of the K-D-E-L signal in mammalian cells. A review of retention signals revealed that the four C-terminal residues can be characterized by the profile polar-polar-E-L. A comparison of this profile with the database (taking less than 0.5 c.p.u. s) gave 22 hits, 18 of which were helical. These data, combined with other information, suggest a retention mechanism where a C-terminal α-helix plays a prominent role.

Conclusions

We have presented a fast database retrieval system, SCAN3D, tailored to the specific problems of integrated protein structure and sequence analysis. The system uses relational retrieval techniques, but is more efficient. Much of the overhead associated with standard relational systems could be avoided for two reasons. The data in the database are perfectly ordered so that no sorting has to take place and no modifications of the database are required so that provisions for deleting, altering or adding are not necessary. Developing our own application-specific database management system rather than employing an existing general relational database management system also enabled us to incorporate options that otherwise would have been unavailable. In protein engineering projects analysing amino acids in their specific environment is much more useful than inspecting individual residues, but general relational database systems cannot easily and efficiently handle relations between individual entries. Therefore, a new method of using the relational paradigm for the analysis of protein structure and sequence characteristics was introduced. The method is based on the comparison of property profiles with all stretches of residues in the database. The examples show that this method works well and that non-trivial information can be retrieved easily. The c.p.u. requirements are also quite modest so that SCAN3D can be used interactively even on a personal computer.

SCAN3D is in continuous development. In subsquent versions statistical analysis of groups of hits (GOHs) will be improved, fuzzy logic will be available as an alternative to allowing a fixed number of mismatches and disk space requirements and execution speed will be optimized. SCAN3Ds features and flexibility make it very suitable for use in protein engineering and design projects. Combined with the many other modules of the WHAT IF program (Vriend, 1990b), it provides an all in one solution for protein sequence, structure and function data analysis. WHAT IF (including SCAN3D) is available from G.V. for a minimal fee; send mail to VRIEND@EMBL-Heidelberg.DE for information.

Acknowledgements

We thank Steve Gardner, Peter Gray and Anna Tramontano for many helpful discussions. We also thank Steve Fuller and Johan Kemmink for providing interesting test cases and our colleagues at Groningen University and EMBL for extensive testing. We thank Roy Omond and his computer group for continuous support.

References

Baumann,G., Frömmel,C. and Sander,C. (1989) *Protein Engng*, **2**, 329–334.
Bernstein,F.C., Koetzle,T.F., Williams,G.J.B., Meyer,E.F.,Jr, Brice,M.D., Rodgers,J.R., Kennard,O., Shimanouchi,T. and Tasumi,M. (1977) *J. Mol. Biol.*, **112**, 535–542.
Bryant,S.H. (1989) *Proteins*, **5**, 233–247.
Burley,S.K. and Petsko,G.A. (1986) *FEBS Lett.*, **203**, 139–143.
Eijsink,V.G.H., Vriend,G., Van den Burg,B., Van der Zee,J.R. and Venema,G. (1992) *Protein Engng*, **5**, 165–170.
Gray,P.M.D., Paton,N.W., Kemp,G.J.L. and Fothergill,J.E. (1990) *Protein Engng*, **3**, 235–243.
Hendrickson,W.A. and Teeter,M.M. (1981) *Nature*, **290**, 107–113.
Hobohm,U., Scharf,M., Schneider,R. and Sander,C. (1992) *Protein Sci.*, **1**, 409–417.
Holm,L. and Sander,C. (1991) *J. Mol. Biol.*, **218**, 183–194.
Huysmans,M., Richelle,J. and Wodak,S.J. (1991) *Proteins*, **11**, 59–76.
Islam,S.A. and Sternberg,M.J.E. (1989) *Protein Engng*, **2**, 431–442.
Jones,T.A. and Thirup,S. (1986) *EMBO J.*, **5**, 819–822.
Kabsch,W. and Sander,C. (1983) *Biopolymers*, **22**, 2577–2637.
Kemmink,J., Van Mierlo,C.P.M., Scheek,R.M. and Creighton,T.E. (1993) *J. Mol. Biol.*, **230**, 312–322.
Lesk,A.M., Boswell,D.R., Lesk,V.I., Lesk,V.E. and Bairoch,A. (1989) *Protein Seq. Data Anal.*, **2**, 295–308.
Levitt,M. (1992) *J. Mol. Biol.*, **226**, 507–533.
Levitt,M. and Perutz,M.F. (1988) *J. Mol. Biol.*, **201**, 751–754.
McGregor,M.J., Islam,S.A. and Sternberg,M.J.E. (1987) *J. Mol. Biol.*, **198**, 295–310.
Morris,A.L., MacArthur,M.W., Hutchinson,E.G. and Thornton,J.M. (1992) *Proteins*, **12**, 345–364.
Munro,S. and Pelham,H.R.B. (1987) *Cell*, **48**, 899–907.
Parsaye,K., Chignell,M., Khoshafian,S. and Wong,H. (1989) *Intelligent Databases. Object-oriented, Deductive Hypermedia Technologies*. Wiley & Sons, New York.
Ponder,J.W. and Richards,F.M. (1987) *J. Mol. Biol.*, **193**, 775–791.
Richardson,J.S. and Richardson,D.C. (1988) *Science*, **240**, 1648–1652.
Sander,C. and Schneider,R. (1991) *Proteins*, **9**, 56–68.
Singh,J. and Thornton,J.M. (1990) *J. Mol. Biol.*, **211**, 595–615.
Vriend,G. (1990a) *Protein Engng*, **4**, 221–223.
Vriend,G. (1990b) *J. Mol. Graph.*, **8**, 52–56.
Vriend,G. and Sander,C. (1991) *Proteins*, **11**, 52–58.

Received April 20, 1993; revised July 6, 1993; accepted August 31, 1993