ELSEVIER

Rapid retrieval of protein structures from databases

Zeyar Aung[1] and Kian-Lee Tan[2]

[1] Institute for Infocomm Research, 21 Heng Mui Keng Terrace, Singapore 119613, Singapore
[2] Department of Computer Science, National University of Singapore, 3 Science Drive 2, Singapore 117543, Singapore

As protein databases continue to grow in size, exhaustive search methods that compare a query structure against every database structure can no longer provide satisfactory performance. Instead, the filter-and-refine paradigm offers an efficient alternative to database search without compromising the accuracy of the answers. In this paradigm, protein structures are represented in an abstract form. During querying, based on the abstract representations, the filtering phase prunes away dissimilar structures quickly so that only a small collection of promising structures are examined using a detailed structure alignment technique in the refinement phase. This article reviews mainly techniques developed for the filtering phase.

Introduction

In structural bioinformatics, 3D structural comparison and structural database searching of proteins play very important roles. For example, we may search an unknown protein against a database of functionally annotated proteins to infer its functions from those found to be structurally similar to it. As another example, we may also search an important structural motif through a protein structure database so as to retrieve the proteins that contain this motif. Structural database searching has many applications in the area of drug discovery. It can be used to verify the 3D structure of a drug target modeled by structural prediction [1]. It can be used to identify the similar structural folds and families unique to pathogenic organisms to select good drug targets [2], etc.

In recent years, advances in laboratory methods (such as MNR and X-ray crystallography) have contributed to a significant increase in the number of known protein 3D structures. For example, the Protein Structure Data Bank (PDB) [3] stored only about 1000 structures in 1993. However, as of August 2007 it stores over 45,000 structures. When the databases were small, exhaustively searching a database by comparing the query structure against each and every structure in the database was done with acceptable performance. However, for large databases with tens of thousands of structures, such an exhaustive searching approach no longer provides a satisfactory response time. As such, much research has gone into developing faster searching algorithms. In particular, we can view these schemes as belonging to the filter-and-refine paradigm.

Figure 1 illustrates the filter-and-refine paradigm. As seen in the schematic drawing, the system introduces a preprocessing phase to map each 3D protein structure into an abstract form (e.g., sequence of symbols or vectors) that can be manipulated more efficiently, for example, comparing two proteins over their abstract representations is much faster than comparing their 3D structures. As a result, besides the original 3D protein structures, a database of abstract representations is also maintained. In some systems, additional data structures called indexes (such as suffix tree, hash table, and inverted file) are created to facilitate speedy access to the abstract representations. During querying, the query structure is first transformed into its abstract representation (using the same method used to map each object in the database). The query answers are then returned in two phases:

1. In the filtering phase, based on the query abstractions, the database of abstract representations is searched. We note that this also requires a similarity measure to be defined for comparing abstract representations. In this way, we can quickly pick out proteins that are similar (under the abstract

Corresponding author: Aung, Z. (azeyar@i2r.a-star.edu.sg), Tan, K.-L. (tankl@comp.nus.edu.sg)

1359-6446/06/$ - see front matter doi:10.1016/j.drudis.2007.07.014

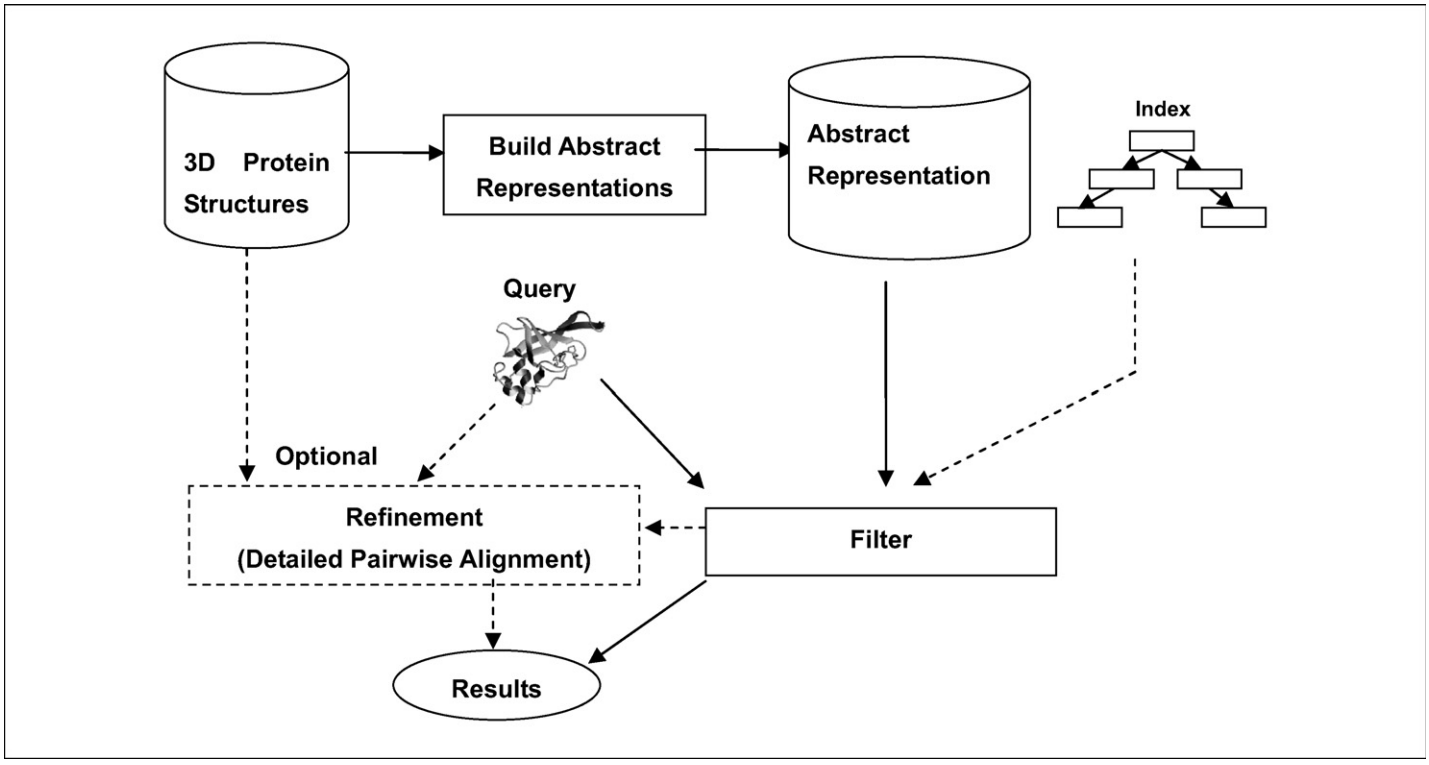

FIGURE 1

The filter-and-refine paradigm for database search.

representations) to the query protein, while pruning away those that are dissimilar. This filtering step can be done in two ways: scanning all the abstract representations in the database or exploiting the precomputed index structure to speed up the retrieval process.

2. In the refinement phase, the potential answers identified in the filtering step are refined by performing a detailed structure alignment comparison between the original query structure and each of the potential answers' structure. Some of the more popular detailed structure alignment schemes include SSAP [4], COMPARER [5], DALI [6], VAST [7], STRUCTAL [8], LSQMAN [9], LOCK [10], CE [11], and FATCAT [12].

By giving priority to speed, rapid search methods employed in the filtering phase generally can offer only a moderate level of accuracy when compared to detailed alignment methods. In other words, the ranking of the answers may not be the same as that of a detailed alignment scheme used in the refinement phase. In addition, some of the rapid search methods such as [13–15] provide only the overall similarity scores of a query protein with respect to the database proteins, but not the detailed residue–residue alignments of the query and database proteins. However, by utilizing the speed advantage of a rapid search method in the filtering phase and the accuracy advantage of a detailed alignment method at the refinement phase, we can achieve a desirable outcome: a fast yet accurate structural database searching.

As an indication of the power of the filter-and-refine strategy, let us compare the performance of CE [11] (a detailed structure comparison method) and Topscan [13] (a fast filter scheme based on abstract representations) on a standard stand-alone Pentium IV PC. CE takes 20 s to compare two proteins, and 800,000 s (nine days!) to search through the entire PDB database with 40,000 proteins. On the contrary, Topscan requires only 0.025 s to compare two proteins and 17 min to search through 40,000 structures. Assuming 1000 of the top ranked proteins from the results of Topscan are selected, the refinement process will take another 20,000 s, or five hours to find the most similar matches. We note that as long as the answers returned by CE are within the top 1000 answers of Topscan, we will have a significant reduction in computational cost without sacrificing the accuracy.

In this review, we will look at rapid search techniques developed for the filtering phase as this phase is particularly critical for the overall speed of the filter-and-refine database search. Moreover, there have been a number of survey papers and books that review and/or evaluate various structural comparison techniques that can be used in the refinement phase [16–22]. This review focuses on fast techniques, and incorporates several recent works that have not been previously reported. We will look at the abstractions that have been used, the indexing methods designed to speed up retrieval, and the similarity measures used in the abstract representations. Box 1 summarizes some common abstract representations and concepts that we will use throughout this paper.

String-based methods

Several efforts have attempted to represent 3D protein structures as strings (sequence of symbols). In this way, the well-established sequence alignment algorithms can be tapped to compare the proteins efficiently.

Secondary structure elements (SSEs)

In Topscan [13], the protein structure is represented as a sequence of SSEs. To further increase the information content, each helix/sheet is encoded with a symbol based on properties such as SSE

BOX 1

Secondary structure element (SSE)

A protein structure can also be characterized by its secondary structure which is the general 3D form of local segments, called secondary structure elements (SSEs), of the protein.
The α-helix and β-strand/sheet are the two most common types of SSEs. The 3_{10}, π, and left-handed helices are special types of helices which are typically treated as an α-helix [10]. The annotation of SSEs is somewhat subjective. Most researchers use DSSP [23] or STRIDE [24] as annotation tools. Both tools have been shown to agree in their SSE annotations in 95% of the cases [25]. An SSE is often modeled as a vector with length and direction obtained from the N- and C-terminal C_α atoms. The main advantage of using SSEs is that the size of a SSE-based structure is significantly smaller than the residue-based structure.

Angles
Several types of angles have also been widely used in protein database search. Consider five residues $i-2$, $i-1$, i, $i+1$, and $i+2$. The κ-angle, ranging from 0° to 180° of a residue i is defined as a bond angle formed by three C_α atoms of residues $i-2$, i, and $i+2$. The α-angle, ranging from −180° to 180° of a residue i is a dihedral (torsion) angle formed by four C_α atoms of residues $i-1$, i, $i+1$, and $i+2$. Angles between SSE vectors can also be defined similarly. Angles are advantageous as they are invariant to translation and rotation of the protein structure in the actual coordinate system.

Distance matrix
A 3D protein structure can also be represented as a 2D distance matrix. The distance matrix (DM) of a protein A with |A| residues is an $|A| \times |A|$ matrix. The (i, j)-entry, DM(i, j), stores the interatomic distance d_{ij} between the two C_α atoms i and j ($1 \leq i, j \leq |A|$) of the protein. The interatomic distance is typically defined by the Euclidean distance between the two atoms. The DM representation is attractive because it is rotation and translation invariant, yet at the same time it captures the same structural information as a 3D representation (and can be used to construct the original 3D form [26]).

Contact pattern
A contact pattern is a submatrix of the distance matrix that captures the interatomic distances between the C_α atoms of two SSEs. There are two types of contact patterns. The intrastructural contact pattern reflects the distance between the C_α atoms of every pair of residues of a single SSE. On the other hand, the interstructural contact pattern captures the distance between the C_α atoms of two different SSEs.

Index structures
To speed up query processing, several data structures can be used. For strings, the suffix tree is commonly used [27]. The suffix tree for a string S is a tree whose edges are labeled with strings, and such that each suffix of S corresponds to exactly one path from the tree's root to a leaf. It is thus a radix tree for the suffixes of S. Once constructed, locating a subsequence in S, locating a substring if a certain number of mistakes are allowed, and locating matches for a regular expression pattern can be performed efficiently.
Where proteins are represented as multidimensional points, then point access methods are typically used [28].
Another structure is the inverted index (or inverted file) [29]. An inverted file stores mapping from words to their locations in a document or a set of documents, allowing full text search. By treating a pattern as a word, and a protein structure as a document, the inverted file can be readily adapted for biological domain.

Similarity measures
Different structural comparison and database search methods use different criteria to measure the similarity between two protein structures [17]. Usually, the similarity measure is dependent on the abstraction used to represent protein structures and the comparison method itself. For example, if a protein structure is represented as a multidimensional vector, the similarity between two proteins is calculated based on the similarity/distance (e.g., Euclidean, Cosine, Pearson, etc.) between two vectors. The validity of a comparison method and its similarity measure can be tested by benchmarking the results against the gold standard databases such as SCOP [30] and CATH [31], or by the standard alignment quality criteria such as root mean square deviation (RMSD).

type (helix, sheet), directions (up, down, left, right, backward, forward), accessibility, proximity, length (short, long), and loop length. For example, to capture only the directional information, Topscan uses 12 symbols (6 for helix, 6 for sheet); if the length information is further added, it will need 24 symbols. Topscan can then apply the Needleman and Wunsch dynamic programming algorithm [32] to align two linear symbolic strings. To compute the similarity between two strings, Topscan also defines the similarity between symbols using a scoring matrix similar to the PAM and BLOSUM matrices used for protein sequence alignment. The matrices, however, are obtained from empirical study conducted over some specific datasets. For each pairwise comparison, Topscan requires a total of 24 alignments to permute the different directions and orientations of the axes. The one that gives the best alignment result is then selected as the answer. While this improves its accuracy, it also led to Topscan's being inferior to (slower than) recently proposed techniques (e.g., SCALE [33] and ProtDex2 [15]).

Five-residue-long structure fragment
Yang and Tung proposed 3D-BLAST for protein structure database search [34]. 3D-BLAST represents a structure as a sequence of symbols, each of which encodes a five-residue-long structure fragment. In 3D-BLAST, there are 23 distinct symbols. These are obtained using a large collection of structurally similar protein pairs with low sequence identity as follows. (a) For each structure, a set of five-residue-long fragments (with residues $i-2$, $i-1$, i, $i+1$, and $i+2$) are extracted. (b) For each fragment, the (κ, α)-pair angle is obtained. (c) The (κ, α)-pair angles are then clustered into 23 clusters. These clusters represent the pattern profiles of the backbone fragments of a protein. To facilitate similarity score computation, a 23×23 substitution matrix is defined based on the observed and estimated probability of occurrences of each (i, j) pair. A database search can then be performed by BLASTing (using BLAST) over the encoded database. 3D-BLAST is the first scheme to provide a function analogous to the *E*-value of BLAST to examine the statistical significance of an alignment hit.

Geometric property
In TLOCAL [35] the key concept is the writhing number which originated from the integral formulas of Vassiliev Knot invariants. Given a protein structure, a sliding window is applied on the protein chain to obtain a sequence of n consecutive α-carbons. The writhing number for each such n consecutive α-carbons is then determined. Based on the distributions of writhing numbers

obtained from a collection of protein structures, the histogram is partitioned into 20 bins and one alphabet is assigned to each bin. To ensure that each alphabet captures approximately the same fraction of observed writhing numbers, the manner of partitioning maximizes the information content as defined by Shannon's information entropy. Each writhing number is thus encoded by an alphabet, and a protein structure by a sequence of alphabets. A substitution matrix was also derived to facilitate comparison of structures encoded by the geometric alphabets.

SSEs + contact patterns

Structure conscious alignment of secondary structure elements (SCALE) [33] adopts SSE as the underlying representation. However, it computes the similarity between two matching subsequences of SSEs based on a matrix of SSE-vector-based dihedral angles and distances (between midpoints of SSE vectors).

To speed up processing, it constructs a hierarchical index of SSE triplets with three levels: (1) nodes with SSE triplet type (α α α, α α β, etc.), (2) nodes with two SSE–SSE dihedral angle ranges, and (3) nodes with two SSE–SSE distance ranges. Each third-level node points to a leaf page containing the protein IDs in which the corresponding triplets occur. When evaluating a query, the index is used to find the IDs of the proteins containing a large enough number of matching SSE triplets with respect to those in the query. These candidate proteins are refined with a dynamic programming-based SSE alignment algorithm, using the dihedral angle and the distance properties of each SSE pair, and a scoring function based on maximally common subsequence (MCS).

SSE triplets + angle

PROuST [36] also operates at SSE level, and attempts to align two proteins over their SSE representations. However, to speed up processing, it creates an inverted file as follows. For a structure with n SSEs, say $(p_1, p_2, \ldots, p_n)$, all valid SSE triplets are obtained. A SSE triplet (p_u, p_v, p_z) is valid if $u < v < z$. For each valid SSE triplet (p_u, p_v, p_z), the angles p_{uv}, p_{vz}, and p_{uz} are obtained where p_{ij} denotes the angle between SSE vectors p_i and p_j. Similarly, three distances (based on midpoints of the SSE vectors) can be obtained between the three SSE vectors of a triplet. The three angles of an SSE triplet and an additional triplet type form a 4D key to proteins in the database that has such a triplet pattern. The triplet type indicates the composition of the triplets in terms of number and position of helices and strands, and is used to resolve the ambiguity of comparing a segment representing a helix and one representing a strand. The distance information is used to further prune away proteins that share similar triplet angles but have different triplet distance. In this way, only protein structures with matching triplets (in terms of both angles and distance) as the query proteins need to be aligned.

Protein substructure

Protein Structure Indexing using Suffix Trees (PSIST) [37] is a string-based indexing method. The relationship between a pair of residues is defined by the distance d between their C_α atoms and the angle θ between the normals to the triangles of N–C_α–C atoms of the residues. Thus, a fragment of size w of a protein's backbone can be regarded as a vector with $2(w-1)$ dimensions: $(d_1, \theta_1, d_2, \theta_2, \ldots, d_{w-1}, \theta_{w-1})$. The attributes in this vector are normalized (discretized) to generate a structure-feature represented by an integer symbol. Thus, a protein structure can be represented as a structure-feature sequence (SF-sequence) with $n - w + 1$ symbols. The SF-sequences of all proteins in the database are used to construct a generalized suffix tree (GST). Given a query and a feature distance threshold ε, the method first extracts the SF-sequence for the query, and then performs searching, ranking and postprocessing. The searching phase traverses the GST to retrieve all the matching segments/subsequences from the database within a distance threshold ε per symbol. The ranking phase ranks all the proteins by chaining the matching segments. The postprocessing step further uses the Smith–Waterman sequence alignment algorithm [38] to find the best local alignment between the query and the selected proteins. The experimental results show that PSIST produced more accurate structural classification results than the other methods such as geometric hashing [39] and PSI [40].

Angles

Polypeptide Angle Suffix Tree (PAST) [41] is another suffix tree-based structural database searching scheme that employs fast string matching. It represents a protein structure as a sequence of angles, given by the dihedral (torsion) angle α. These angles are further encoded into an alphabet by discretizing in intervals of size $360^\circ/36 = 10^\circ$. To speed up the retrieval process, PAST employs a suffix tree to index the sequence of angle alphabets. In addition, both exact match and approximate search are supported. For approximate search, search paths involving neighboring characters (corresponding to neighboring angles) are also examined.

Component-based vector methods

Recently, attempts have been undertaken to represent protein structures as vectors (i.e., a multidimensional point). In this way, multidimensional indexing structures can be employed to speed up the retrieval process. In this section, we look at methods where the vectors are derived from the components (substructures) of a protein.

Contact patterns

Protein Indexing (ProtDex2) [15] extracts contact patterns of SSEs from a 3D structure. The intuition is that proteins with similar contact patterns are likely to be structurally similar. Since contact patterns vary in size, ProtDex2 generates a set of fixed-size matrices from each contact pattern by applying a fixed-size sliding window over the contact pattern. Therefore, each protein structure is represented by a set of such fixed-size matrices. To further compact the representation, instead of storing the matrices, ProtDex2 stores each matrix as a d-dimensional feature vector $(v_1, v_2, \ldots, v_d)$ where each v_1 approximates some statistical information of the matrix. Statistics used in ProtDex2 include the angle and distance between two SSE vectors, the mean and standard deviation of the C_α–C_α distance in the contact pattern, and so on. In this way, structure comparison is performed based on the feature vector. The similarity measure adopted is a weighted sum of the similarity between each component statistical information. To speed up the query processing, an inverted file is constructed – from each vector, we can determine the list of structures from which such a vector can be extracted. Given a query structure, its set of feature vectors is first extracted. The inverted index is then searched to pick out

those relevant structures. This search process is done quickly using a hash table to locate matching vectors. The set of proteins can then be ranked based on the similarity measure. Thus, only proteins that are similar will be retrieved, while dissimilar proteins are pruned off when the inverted file is searched.

Protein substructure

Recently, Huang *et al.* developed a method to represent a protein by its substructure, called patches [42]. The building block of a patch is a bowtie, which comprises two vectors v_i and v_j where v_i and v_j are vectors obtained from the C_α and C_β atoms of residues i and j, respectively, and the distance between the C_α atoms of the two vectors are within a certain threshold distance (15 Å is used). A patch is a set of vectors such that all pairs of vectors in this set are bowties. By fixing the size of the patch to k, a patch can be represented as a ($7k$-10)-vector. Hence a dimensional index can be used to support retrieval. The scheme preprocesses the database of structures to generate all patches. This way, a query is evaluated by searching for matching patches. As the dimensionality of the vector is very high, the dimensionality is further reduced using Singular Value Decomposition [43] and Locality Preserving Projection [44].

SSE triplets

Protein Structure Indexing (PSI) [40] represents a protein structure as set of SSE triplets. Each triplet is encoded as a feature vector in a six-dimensional space, describing three distances and three angles among the SSEs in the triplet. Such feature vectors are extracted from all protein structures in the database, and an R*-tree is built on this feature space using Minimum Bounding Rectangles (MBRs). When evaluating a query, the index is searched to quickly find the database protein's triplets that match those of the query. Each matching SSE triplet pair is assigned a score based on inverse root mean square deviation (RMSD) of the corresponding SSEs. Then, the method constructs a triplet pair graph (TPG) with its vertices corresponding to the aligned triplet pairs. An edge is placed between two vertices if they share two SSE mappings. A depth first search algorithm is used on the TPG to find the largest weighted connected component (LWCC). The LWCC of the TPG corresponds to the most similar subset of SSEs of database proteins and the query SSEs. Then, a bipartite graph on LWCC is constructed with one set of vertices being the database protein's SSEs, and the other set being the query's SSEs. The weight of an edge shows how good the alignment is between the corresponding vertices. Then, the method applies a largest weighted bipartite graph matching to find the seed alignment of the SSEs. The significance of each database protein's seed alignment is evaluated by a P value statistical model. Then, the detailed refinement using the VAST method [7] is carried out on the database proteins whose seed alignments are significant. PSI is reported to have improved the pruning time of VAST 3–3.5 times while maintaining similar sensitivity.

Structure-based vector methods

Unlike the component-based methods, under the structure-based methods, the vectors are obtained from the entirety of the protein structure. As such, the number of dimensions of the vector is typically much higher.

Distance matrix

Protein database search (ProteinDBS) [45] is an image processing-based method that exploits information stored in the 2D distance matrix. It partitions each distance matrix into four diagonal bands, and constructs distance histograms with the bin width of 10 Å. A number of texture attributes, in particular energy, entropy, homogeneity, contract, correlation, and cluster tendency are also computed from the distance matrix. The method maps a protein structure into a multidimensional point whose features are the histogram bin values and the texture attribute values. It stores the multidimensional points representing the protein structures in the database as an Entropy Balanced Statistical (EBS) k-d tree. The EBS k-d tree is trained with a selected set of points (a subset of the entire database) with the known SCOP [30] protein structure classification labels. The dimensionality of the points in the tree is reduced by determining the discriminant features of the training (labeled) points. This results in 23-dimensional points indexed in the EBS k-d tree. A partial clustering based on the training points is carried out to obtain the optimal structure of the tree. In evaluating a query, a binary search is performed on the tree and finally the leaf pages that contain the IDs of relevant proteins are returned as the answer. The answer proteins are sorted based on their k-dimensional similarity to the query. The method is reported to be very fast: it can query a database of 46,075 protein structures in less than 10 s.

Projection-based vector methods

In projection-based methods, a database of 3D protein structures is mined to extract p representative patterns. A protein structure can then be represented by a p-vector whose i-dimension captures the weighted number of times the pattern is observed in the structure. As such, two protein structures are similar if they share similar number of each pattern. This can be determined very efficiently using distance metric between the corresponding vector representations of the structures (e.g., Euclidean, Pearson). While projection methods are essentially vector-based, they differ from vector-based methods as they only exploit a subset of representative patterns.

In [46], a projection method is proposed. There, a triplet of SSEs is adopted as the pattern that models the structural fragment. Essentially, each SSE is approximated by a positional vector in 3D and the spatial conformation of an SSE triplet is represented by the relative orientation of the corresponding SSE vectors (i.e., all pairwise angles and distances between the midpoints of the corresponding SSE vectors). Using a representative set of protein structures in the database, all their SSE triplets are extracted. These SSE triplets are then organized into eight groups based on their SSE types ($\alpha\ \alpha\ \alpha$, $\alpha\ \alpha\ \beta$, …, $\beta\ \beta\ \beta$). The K-mean clustering algorithm is then used to cluster the triplets within each group. In total, p clusters are generated, and hence p representative patterns (the cluster centers) are derived. Now, for each database protein, its SSE triplets are extracted, and each triplet is then assigned to one of the p patterns. The protein is then represented as a p-vector, where the ith dimension denotes the weighted number of SSE triplets that is similar to the ith pattern. Two protein structures are similar if they have matching vectors. In the work, the distance between two vectors is measured using the Pearson correlation coefficient.

The effectiveness of projection methods depends on the number of patterns/clusters that are used as models. Moreover, new

structural fragments (or SSE triplets) derived from newly discovered proteins may not be represented unless the models are recomputed.

Histogram-based methods

Probability of identity (PRIDE) is a histogram-based scheme whose histograms are obtained from the distance matrix of a protein [14]. It defines a residue position difference n, and calculates all the distances between residues i and $i+n$ to construct a distance histogram. In this way, given a particular n value, it can generate a histogram on the distribution of distances between residues that are n atoms apart. The width for each bin in the histogram is set to 0.5 Å. The probability of identity (PRIDE) score of two histograms (representing two protein structures) is assessed by contingency-table analysis based on χ^2 test. The χ^2 value is calculated based on the distributions of the values in the m number of bins of the two histograms. The probability of the two distributions being identical is read from the corresponding χ^2 distribution of $m-1$ degrees of freedom. In the method, 28 histogram pairs (with $n = 3$–30) are constructed, and the 28 PRIDE scores are averaged. The PRIDE score is reported to observe the metric properties (non-negativity, identity, symmetry, and triangular inequality) in most of the cases, and be highly correlated to the conventional RMSD measure. PRIDE is an example of nonalignment-based structural comparison method. The method is reported to be very fast.

Graph-based representation

It is also common to find searching techniques that represent structures as graphs. As graphs have been widely studied in the computing/mathematics literature, many known properties can be exploited to speed up the retrieval process.

Residues

k-Clique hashing [47] exploits both the accuracy advantage of maximum clique detection-based techniques and the efficiency of geometric hashing techniques. The method represents a protein structure as a graph with its residues as nodes. If the distance between two nodes is less than 12 Å, an edge is inserted between them. Each node is labeled with the physico-chemical properties of the residue center, and each edge is labeled with the distance between the residues. Protein structure comparison using graph representation generally applies maximum clique (complete subgraph) detection in the node-product graph of the two input graphs in order to find the largest common isomorphic subgraph of them [7,48,49]. Maximum clique detection is very compute-intensive with most of the time being spent on the elimination of false positive matches of nodes and the assembling of real matches into larger matching. The *k*-clique hashing method addresses this problem by introducing larger matches of size *k* which leads to a much simpler node-product graph and a faster assembly process. The method subdivides all graphs for proteins in the database into *k*-cliques ($k = 3$ is used as default); maps them as a point into a Euclidean space; and indexes them using an R*-tree (which can be regarded as a hash table with variable-sized cells based on the points' distribution). When a query is evaluated, the index is used to find the matching *k*-cliques of the database proteins to those of the query. The matching *k*-cliques for each database protein are assembled into a larger clique by combining the overlapping *k*-cliques. To reduce the number of false positives in this assembly step, a heuristics (called hits list voting) is used.

SSE vectors

Secondary Structure Matching (SSM) [50] represents a protein as a complete graph of SSE vectors. Each node is an SSE vector, and each edge between two nodes represents certain information between the two SSEs. These include two angles, two torsion angles, distance, and connectivity information. The corresponding SSEs between the two graphs, representing two protein structures, are identified by a rapid common subgraph isomorphism algorithm called CSIA with the time complexity of $O(m^{n+1}n)$, where m and n are the number of nodes (SSEs) in two graphs. This algorithm is much faster than the conventional algorithm

TABLE 1

List of rapid protein structure database retrieval methods described in this review

Method	Year of publication	References	How to access/obtain
Topscan	2000	[13]	http://www.bioinf.org.uk/topscan/
3D-BLAST	2006	[34]	http://3d-blast.life.nctu.edu.tw/
TLOCAL	2006	[35]	Contact T. Gregory Dewey (greg_dewey@kgi.edu)
SCALE	2005	[33]	Contact K.-L. Tan (tankl@comp.nus.edu.sg)
PROuST	2004	[36]	Please see reference
PSIST	2005	[37]	Please see reference
PAST	2006	[41]	http://past.in.tum.de/
ProtDex2	2004	[15]	http://www1.i2r.a-star.edu.sg/~azeyar/genesis/ProtDex2/
Huang *et al.*	2006	[42]	Please see reference
PSI	2004	[40]	http://bioserver.cs.ucsb.edu/proteinstructuresimilarity.php
ProteinDBS	2004	[45]	http://proteindbs.rnet.missouri.edu/
Zotenko *et al.*	2006	[46]	http://www.biomedcentral.com/content/supplementary/1472-6807-6-12-s12.tar
PRIDE	2002	[14]	http://hydra.icgeb.trieste.it/pride/
k-Clique Hashing	2004	[47]	Contact E. Hullermeier (eyke@mathematik.uni-marburg.de)
SSM	2004	[50]	http://www.ebi.ac.uk/msd-srv/ssm/

with the complexity $O[(mn)^n]$. Then, the initial alignment of the corresponding SSEs is carried out by a fast optimal superposition procedure. This initial alignment is iteratively refined and expanded into the final one (at the residue level) by a combination of five techniques: (1) mapping C_α atoms of matched SSEs, (2) mapping C_α atoms of the nonmatched SSEs, (3) expansion of C_α atom contacts, (4) quality filter using a quality scoring function based on RMSD and the number of aligned residues, and (5) unmapping short fragments. The significance of the alignment is evaluated with statistical means, viz., *P* value and *Z* score. SSM is reported to be quite fast and pretty accurate [19].

Conclusion

In this review, we have looked at techniques that have been developed to speed up structure database search. The basic approach is to represent protein structures in an abstract form that is more computationally efficient to process. Once potentially similar proteins have been identified, a detailed structure alignment scheme can be applied to further refine the answers. Many of these schemes have been shown to be efficient, without sacrificing the accuracy in comparison to the detailed structure alignment techniques. Table 1 summarizes the techniques that we have discussed.

Due to space limitations, this review is by no means exhaustive. Other methods worthy of mention include: structure shape-based method [51], wavelet-based method [52], graph-based methods [48,49,53,54], geometric hashing schemes [39,55], cartoon representation [56], SSE–SSE interaction matrices [57], probabilistic SSE matching [58], topological representation [59], and residue pairing [60].

It should be noted that the area of index-based structural database searching is relatively immature and none of the methods is known to be widely used at present. We hope this review will prompt biologists and researchers in the drug discovery community to find these techniques useful in their work.

References

1 Wieman, H. *et al.* (2004) Homology-based modelling of targets for rational drug design. *Mini Rev. Med. Chem.* 4, 793–804
2 Gerstein, M. (2000) Integrative database analysis in structural genomics. *Nat. Struct. Biol.* 7 (Suppl.), 960–963
3 Berman, H.M. *et al.* (2000) The Protein Data Bank. *Nucleic Acids Res.* 28, 235–242
4 Taylor, W.R. and Orengo, C.A. (1989) Protein structure alignment. *J. Mol. Biol.* 208, 1–22
5 Sali, A. and Blundell, T.L. (1990) Definition of general topological equivalence in protein structures: a procedure involving comparison of properties and relationships through simulated annealing and dynamic programming. *J. Mol. Biol.* 212, 403–428
6 Holm, L. and Sander, C. (1993) Protein structure comparison by alignment of distance matrices. *J. Mol. Biol.* 233, 123–138
7 Gibrat, J.F. *et al.* (1996) Surprising similarities in structure comparison. *Curr. Opin. Struct. Biol.* 6, 377–385
8 Levitt, M. and Gerstein, M. (1998) A unified statistical framework for sequence comparison and structure comparison. *Proc. Natl. Acad. Sci. U.S.A.* 95, 5913–5920
9 Kleywegt, G.J. (1996) Use of non-crystallographic symmetry in protein structure refinement. *Acta Cryst. Sect. D* 52, 842–857
10 Singh, A.P. and Brutlag, D.L. (1997) Hierarchical protein structure superposition using both secondary structure and atomic representations. In *Proceedings of the Fifth International Conference on Intelligent Systems for Molecular Biology* pp. 284–293
11 Shindyalov, I.N. and Bourne, P.E. (1998) Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. *Protein Eng.* 11, 739–747
12 Ye, Y. and Godzik, A. (2003) Flexible structure alignment by chaining aligned fragment pairs allowing twists. *Bioinformatics* 19 (Suppl. 1), 246–255
13 Martin, A.C.R. (2000) The ups and downs of protein topology: rapid comparison of protein structure. *Protein Eng.* 13, 829–837
14 Carugo, O. and Pongor, S. (2002) Protein fold similarity estimated by a probabilistic approach based on C_α–C_α distance comparison. *J. Mol. Biol.* 315, 887–898
15 Aung, Z. and Tan, K.L. (2004) Rapid 3D protein structure database searching using information retrieval techniques. *Bioinformatics* 20, 1045–1052
16 Carugo, O. (2006) Rapid methods for comparing protein structures and scanning structure databases. *Curr. Bioinform.* 1, 75–83
17 Eidhammer, I. *et al.* (2000) Protein structure comparison and structure patterns. *J. Comput. Biol.* 7, 685–716
18 Koehl, P. (2001) Protein structure similarities. *Curr. Opin. Struct. Biol.* 11, 348–353
19 Kolodny, R. *et al.* (2005) Comprehensive evaluation of protein structure alignment methods: scoring by geometric measures. *J. Mol. Biol.* 346, 1173–1188
20 Lancia, G. and Istrail, S. (2003) Protein structure comparison: algorithms and applications. In *Mathematical Methods for Protein Structure Analysis and Design* (Guerra, C. and Istrail, S., eds), pp. 1–33, Springer-Verlag
21 Novotny, M. *et al.* (2004) Evaluation of protein fold comparison servers. *Proteins: Struct. Funct. Bioinform.* 54, 260–270
22 Sierk, M.L. and Pearson, W.R. (2004) Sensitivity and selectivity in protein structure comparison. *Protein Sci.* 13, 773–785
23 Kabsch, W. and Sander, C. (1983) DSSP: definition of secondary structure of proteins given a set of 3D coordinates. *Biopolymers* 22, 2577–2637
24 Frishman, D. and Argos, P. (1995) Knowledge-based secondary structure assignment. *Proteins: Struct. Funct. Genet.* 23, 566–579
25 Martin, J. *et al.* (2005) Protein secondary structure assignment revisited: a detailed analysis of different assignment methods. *BMC Struct. Biol.* 5:17
26 Crippen, G.M. and Havel, T.F. (1988) *Distance Geometry and Molecular Conformation*. John Wiley & Sons
27 Hunt, E. (2001) A database index to large biological sequences. In *Proceedings of the 2001 Very Large Data Base Conference* pp. 139–148
28 Samet, H. (1990) *The Design and Analysis of Spatial Data Structures*. Addison-Wesley
29 Frakes, W.B. and Baeza-Yates, R.A. (1992) *Information Retrieval: Data Structures and Algorithms*. Prentice-Hall
30 Hubbard, T.J.P. *et al.* (1997) SCOP: a structural classification of proteins database. *Nucleic Acids Res.* 25, 236–239
31 Orengo, C.A. *et al.* (1997) CATH: a hierarchic classification of protein domain structures. *Structure* 5, 1093–1108
32 Needleman, S.B. and Wunsch, C.D. (1971) A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.* 48, 443–453
33 Chionh, C.H. *et al.* (2005) Towards scaleable protein structure comparison and database search. *Int. J. Artif. Intell. Tools* 14, 827–848
34 Yang, J.M. and Tung, C.H. (2006) Protein structure database search and evolutionary classification. *Nucleic Acids Res.* 34, 3646–3659
35 Chang, P.L. *et al.* (2006) Structure alignment based on coding of local geometric measures. *BMC Bioinform.* 7:346
36 Comin, M. (2004) PROuST: a server-based comparison method of three-dimensional structures of proteins using indexing techniques. *J. Comput. Biol.* 11, 1061–1072
37 Gao, F. and Zaki, M.J. (2005) PSIST: indexing protein structures using suffix trees. In *Proceedings of the Fourth IEEE Computational Systems Bioinformatics Conference* pp. 212–222
38 Smith, T.F. and Waterman, M.S. (1981) Comparison of biosequences. *Adv. Appl. Math.* 2, 482–489
39 Nussinov, R. and Wolfson, H.J. (1991) Efficient detection of three-dimensional structural motifs in biological macromolecules by computer vision techniques. *Proc. Natl. Acad. Sci. U.S.A.* 88, 10495–10499
40 Camoglu, O. *et al.* (2004) Index-based similarity search for protein structure databases. *J. Bioinform. Comput. Biol.* 2, 99–126
41 Taubig, H. *et al.* (2006) PAST: fast structure-based searching in the PDB. *Nucleic Acids Res.* 34, W20–W23 web server issue
42 Huang, Z. *et al.* (2006) Dimensionality reduction in patch-signature based protein structure matching. In *Proceedings of the 17th Australiasian Database Conference* pp. 89–97
43 Golub, G. and Van Loan, C. (1996) *Matrix Computations*. John Hopkins University Press

44 He, X. *et al.* (2004) Locality preserving indexing for document representation. In *Proceedings of the 27th ACM SIGIR Conference* pp. 96–103
45 Shyu, C.R. (2004) ProteinDBS – a content-based retrieval system for protein structure database. *Nucleic Acids Res.* 32, 572–575
46 Zotenko, E. *et al.* (2006) Secondary structure spatial conformation footprint: a novel method for fast protein structure comparison and classification. *BMC Struct. Biol.* 6:12
47 Weskamp, N. *et al.* (2004) Efficient similarity search in protein structure databases by k-clique hashing. *Bioinformatics* 20, 1522–1526
48 Grindley, H. *et al.* (1993) Identification of tertiary structure resemblance in proteins using a maximal common sub-graph isomorphism algorithm. *J. Mol. Biol.* 229, 707–721
49 Koch, I. and Lengauer, T. (1997) Detection of distant structural similarities in a set of proteins using a fast graph-based method. In *Proceedings of the Fifth International Conference on Intelligent Systems for Molecular Biology* pp. 167–187
50 Krissinel, E. and Henrick, K. (2004) Secondary-structure matching (SSM), a new tool for fast protein structure alignment in three dimensions. *Acta Cryst. Sect. D* 60, 2256–2268
51 Ankerst, M. *et al.* (1999) 3D shape histograms for similarity search and classification in spatial databases. In *Proceedings of the Sixth International Symposium on Spatial Databases* pp. 207–226
52 Marsolo, K. *et al.* (2006) Structure-based querying of proteins using wavelets. In *Proceedings of the 2006 ACM CIKM International Conference on Information and Knowledge Management* pp. 24–33
53 Taylor, W.R. (2002) Protein structure comparison using bipartite graph matching and its application to protein structure classification. *Mol. Cell. Proteomics* 1, 334–339
54 Harrison, A. *et al.* (2003) Recognizing the fold of a protein structure. *Bioinformatics* 19, 1748–1759
55 Bachar, O. *et al.* (1993) A computer vision based technique for 3-D sequence independent structural comparison of proteins. *Protein Eng.* 6, 279–288
56 Gilbert, D. *et al.* (1999) Motif-based searching in TOPS protein topology databases. *Bioinformatics* 15, 317–326
57 Ohkawa, T. *et al.* (1999) A method of comparing protein structures based on matrix representation of secondary structure pairwise topology. In *Proceedings of the Fourth IEEE Symposium on Intelligence in Neural and Biological System* pp. 10–15
58 Shih, E.S.C. and Hwang, M.J. (2003) Protein structure comparison by probability-based matching of secondary structure elements. *Bioinformatics* 19, 735–741
59 Bostick, D.L. *et al.* (2004) A simple topological representation of protein structure: implications for new, fast, and robust structural classification. *Proteins: Struct. Funct. Bioinform.* 56, 487–501
60 Zhu, J. and Weng, Z. (2005) FAST: a novel protein structure alignment algorithm. *Proteins: Struct. Funct. Bioinform.* 58, 618–627