USING FAST SUBGRAPH ISOMORPHISM CHECKING FOR PROTEIN FUNCTIONAL ANNOTATION USING SCOP AND GENE ONTOLOGY

DEEPAK BANDYOPADHYAY,* JUN HUAN, JINZE LIU, WEI WANG, JAN PRINS, JACK SNOEYINK

Department of Computer Science,
University of North Carolina, Chapel Hill, NC 27599 3175, USA
E-mail: $\{debug, huan, liuj, weiwang, prins, snoeyink\}$*@cs.unc.edu*

We describe a method for protein family identification using a graph representation of proteins. The method incorporates a novel fast subgraph isomorphism method based on a graph index to query a new structure for occurrences of family fingerprints and to assign it to a protein family with a confidence value. This method can provide an independent assignment of the protein family for a new structure *in silico*, in cases where sequence alignments and structural matches fail to provide proper annotation. Using Gene Ontology and cross validation, we further validate the annotation power of the mined fingerprints.

1. Motivation

Structural genomics projects have produced a large number of protein structures which are encoded in the fully-sequenced genomes; the ultimate goal of these projects is to solve structures of all possible protein folds. These projects present a serious challenge [26] for conventional structural biology, as a large and growing number of structures produced have unknown function.

Several functional annotation methods have been proposed, which broadly fall into two categories – sequence-based annotation methods (based on multiple sequence alignment or study of phylogeny and evolution) and structure-based annotation methods. Methods based only on sequences work when they can determine an alignment with another protein or domain of known function with at least 40% sequence identity. Proteins with less than 40% sequence identity (i.e. in the so-called twilight zone of sequence alignment) are not guaranteed to have homologous structures, and can diverge in function even if the structures are similar.

It is widely known that protein function relies on its structure and that structure is better conserved during evolution than sequence [28]. Structure-based annotation

* Portions of this research were supported by NSF grants 0076984 and 9988742.

methods would thus be expected to offer clues about an unknown function; indeed, global structural similarity to a protein of known function often indicates functional similarity. If both sequence and global structure similarity fail to reveal a function, insights from local function-related patterns become critical. Features that could be identified by local structural patterns include the nucleotide-binding surface motif of P-loop containing proteins [31]; the functional core of native Triose Phosphate Isomerases and those redesigned on a Ribose Binding Protein template [7]; and the Ser-His-Asp catalytic triad in all Serine Proteases and several other classes of enzymes such as the $\alpha\beta$-hydrolase and esterase families [6]. Some of these cases are believed to have arisen by convergent evolution.

Patterns such as the catalytic triad, often called residue packing patterns or structural motifs, occur in several different families of proteins. There are sets of patterns that occur together in almost all members of a family, and very rarely in the rest of the PDB. Such a combination of residue packing patterns, called family signatures or fingerprints [13], uniquely identifies a family and thus can be used to decide if a new structure belongs to the family or not.

In this paper, we present a novel automated algorithm to assign a protein a functional family, using local structural patterns which are highly associated with known functional families. Our method works in two stages, built upon our earlier work on deriving protein family-specific fingerprints [13,15,12]. In the first stage, we use a fast subgraph isomorphism algorithm to find all occurrences of family-specific subgraph fingerprints in a protein to be annotated. In the second stage, we assign a family and a significance score to the query, depending on the fingerprints found in it, and search the Gene Ontology (GO) and SCOP family databases to detect functional neighbors of the family.

The rest of this paper is organized into the following sections: the Related work section summarizes current methods for discovering local structural motifs associated with function. The Methods section discusses the techniques used for obtaining and prioritizing family-specific fingerprints, searching for them in a query protein and classifying the protein using the results, and clustering the Gene Ontology. The Experiments section covers the performance improvements from using the graph index, validation of our methods and their application to classify protein families and find functional neighbors of an existing family in SCOP and GO.

1.1. *Related Work*

Traditionally, family assignment was done by global structural alignment or fold comparison. Domain-based methods cluster protein folds by expert human judg-

ment(SCOP [21]); combined expert and automated recognition (CATH [23]); Hidden Markov Models based on sequence [4,10,18] and structure [1]; and structure comparision (DALI [11],VAST [19,9],PRiSM [35]). Recently, attention has shifted to structure similarity at a much more local level than that of fold (domain), where interesting patterns are composed of a limited number of residues. The rationale behind the shift is that the real function is usually carried by a few residues which, if mutated, have significant effect on the protein function.

Comparing to the relatively developed field of protein fold comparison and classification, the algorithms to find local structure motifs are limited. We overview here two major methods; see also recent reviews on methods for assigning function from structure [16,17]. The first method is geometric hashing, originally developed in computer vision, and successfully applied to comparing a pair of protein structures [22] and a protein structure to a structure database [3,32]. Patterns identified by geometric hashing include the serine protease catalytic triad and the ribonuclease and lysozyme catalytic domains [32]. The second method uses subgraph matching to detect recurring structural motifs [2,20,24,25,27,34]. Many patterns are found by this method which include the catalytic triad, a His-His porphyrin binding pattern, and the zinc-finger Cys-Cys-His-His patterns. Compared to geometric hashing, graphs may have labeled edges and nodes and thus in addition to geometric information they can model residue charge, residue-residue interaction, bond type, sequence numbers and other information. Other methods based on Inductive Programming Language [29] and Fuzzy Functional Forms [8] are also used for inferring structure motifs from protein 3D structures.

Our approach for finding local structural patterns is related to methods from graph theory and data mining but with significant improvements. We extend the pair-wise comparison to multiple structure comparison and enhance the algorithm to find patterns of any topologies with arbitrary sizes. Our method assumes no prior knowledge about functional features to be searched for (but such knowledge can be easily incorporated). The method is fully automated and fast enough to find family-specific fingerprints [13] in large families of structurally similar or dissimilar proteins.

2. Methods

2.1. *Modeling Protein Structures by Graphs*

To find recurring substructural motifs, we represent protein structures as labeled undirected graphs, and search for common subgraphs in the structures of the proteins in a family. Our protein graphs have a node for every residue, and pairs of nodes connected by edges of two types: peptide bond edges that connect two primary-sequence consecutive residues, and spatial proximity edges that connect two residues that are nearby in 3D space but not consecutive along the primary sequence. Nodes are labeled by residue type. It is possible to merge two or more node types to create a reduced set of node labels, and to further classify edges based on ranges of edge lengths. We determine spatial proximity between nodes using the almost-Delaunay (AD) edges [13]. An almost-Delaunay graph gives sets of neighbors for each residue in the presence of a bounded uncertainty in the point coordinates, and is parameterized by this uncertainty. While not much larger or slower to compute than Delaunay graphs [13], they are more robust to small changes in the point coordinates.

2.2. *Mining Protein Family Fingerprints*

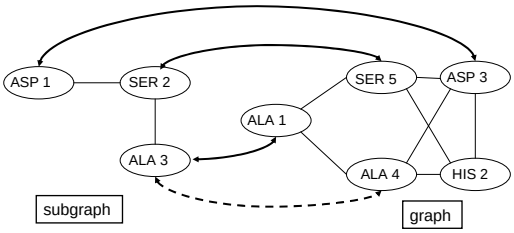
We mine frequent subgraphs from the graph representations of multiple proteins using an algorithm [14] based on a depth-first search on a spanning tree representation of subgraphs; this is much faster than exhaustive enumeration or clique detection. A frequent subgraph is defined as one that occurs in more than a fraction s_F of the proteins in the family; s_F is called the *minimum support* and is by default chosen as 0.9. A *maximal frequent* subgraph has no supergraph that is frequent.

Any subgraphs found to be frequent in the family are then checked against the *background*, a dataset of 6500 non-redundant proteins from CulledPDB [33] with parameters as shown in Table 1, that represents the whole Protein Data Bank. Any subgraph that occurs in more than a fraction s_B of the background is removed from consideration; s_B is denoted the *maximum background frequency* and is by default chosen as 0.05. The remaining frequent subgraphs with high family support and low background frequency are sorted in decreasing order of size and increasing order of background frequency; these correspond to spatial packing patterns that are unique to the family and rarely seen outside of it, and are stored as the family fingerprints.

2.3. *Querying a New Protein*

The problem of annotating the query protein with a set of fingerprints derived from a putative family then becomes one of searching the graph of the query protein for occurrences of subgraphs. Done naively, this is a subgraph isomorphism search, which is known to be an NP-complete problem. However, we use an index of graph similarity to speed up the search.

To quickly filter the subgraphs matching a pattern, we build a k-level local neighborhood index for the graph database of family fingerprints. At each vertex in a graph we store a $20 \times k$ matrix, where entry i, j contains the number of residues of each type i that can be reached from the vertex by a path of length at most j. This is shown in Figure Figure 1. We typically choose $k = 3$ since the subgraphs are typically small (3–12 residues), and since proteins are compact—higher-level indices have less discriminating power. The computation of these numbers is easy using the symmetric adjacency matrix of the graph. If A is the adjacency matrix with ones on the diagonal, then the off-diagonal entries give the adjacencies for level 1, and multiplying the adjacencies for level j by A, setting the diagonal to zero and all positive values to unity gives the adjacencies for level+$(j+1)$.



Node	Label	Neighbor String
1	ASP	*1*: S1 *2*: A1S1
2	SER	*1*: A1D1
3	ALA	*1*: S1 *2*: D1S1

Node	Label	Neighbor String
1	ALA	*1*: A1D1H1S1 2:S1
2	HIS	*1*: A1D1S1 *2*: A2D1S1
3	ASP	*1*: A1H1S1 *2*: A2H1S1
4	ALA	*1*: A1S1 *2*: A1D1H1S1
5	SER	*1*: A2D1H1

Figure 1. We show an example of the proposed graph index. Top: the two matching graphs. A solid arrowed line connects two matching node; the dotted-arrowed line connects two node which have the same type but have different graph index and hence could not be a matching pair. Bottom: the node id, node type, and its index (neighbor string) for each node in the two graphs.

We say that a query matches a subgraph at a vertex by comparing entries of their $20 \times k$ matrices: A query vertex u can match a subgraph vertex v with the same label only if each entry of the index matrix of u is at least the corresponding

entry of the matrix of v. This is illustrated in Figure 1. We can compare the index matrices row by row from shorter to longer paths, and eliminate from consideration all subgraphs that do not have any matches for all their vertices in the query. If a subgraph satisfies the graph index, this does not guarantee an embedding in the query; we subsequently use Ullman's [30] algorithm to verify these matches and find valid occurrences of the subgraph from them.

We define two measures of pruning performance of our graph index: The **efficiency** (η) of a graph index for a particular dataset of query and graph database is the average ratio of the number of true matches for a subgraph in the query and the number of index matches for the same subgraph. Index matches are possible residue assignments that satisfy the graph index, and thus η measures the overhead in pruning non-adjacent embeddings. For the dynamic programming algorithm, we can estimate η for each subgraph as the fraction of nodes traversed that led to successful matches. Dynamic programming is much more efficient than an exhaustive match that may have to enumerate an exponential number of index matches.

Another measure is the **hit rate**, the ratio of the number of subgraphs actually present within a query to the number returned by the graph index, i.e. the number of subgraphs with nonzero match efficiency.

2.4. *Significance of family assignment*

Having found the fingerprints of family F that occur in a query protein q, the simplest method to assign a significance to q being in F is by counting the fingerprints and comparing with the expected counts for family and background proteins, based on support values s_B and s_B used to select fingerprints. If there are C_F fingerprints, an average family protein is expected to have $s_F C_F$ of them, an average background protein to have $s_B C_F$, and both these are normally distributed about their means with variances $C_F s_F(1-s_F)$ and $C_F s_B(1-s_B)$ respectively. These distributions can be used to assign a p-value for the query with N_q fingerprints belonging to the family or the background. This model is crude since it assumes that all fingerprints are equally discriminating and are independent, but it is a good first approximation of the family assignment. A better model is from the joint probability distribution of the fingerprints $X_{q1} \ldots X_{qn}$ found in query q:

$$P(q \in F | X_{q1} \ldots X_{qn} \in q) = \frac{P(q \in F) P(X_{q1} \ldots X_{qn} \in q | q \in F)}{\sum_{S \in \{F,B\}} P(q \in S) P(X_{q1} \ldots X_{qn} \in q | q \in S)}$$

Here we estimate $P(q \in F)$ as N_F/N, where N_F is the number of proteins in the family and N is the total number of proteins in the background.

2.5. *Gene Ontology Enrichment Assessment*

The GO Consortium [5] was formed to integrate the efforts to regulate the vocabulary for various genomic databases of diverse species in such a way that it can show the essential features shared by all the organisms. GO terms and the "is-a" and "part-of" relationships form directed acyclic graphs(DAGs) in which a parent node describes functions exhibited by its child nodes. Terms that are lower in height (i.e. close to the root) describe more general functions; the greater the height, the more specific the function.

The goal of GO enrichment evaluation is to measure whether a set of proteins sharing a fingerprint is enriched with proteins from a particular category to a greater extent than that would be expected by chance. A geometric distribution is used to model the probability of observing at least k proteins from n proteins sharing the same fingerprint by chance in a category containing f proteins from a total protein size of g. The P-value is given by $P = 1 - \sum_{i=0}^{k} \frac{\binom{f}{i}\binom{g-f}{n-i}}{\binom{g}{n}}$. For example, if the majority of proteins in the list appear from one category, then it is unlikely that this happens by chance and the category's P-value would be close to 0.

3. Experiments and Results

3.1. *Performance of the Graph Index*

All the family datasets used in our experiments, along with the number of fingerprints found, are listed in Table 1.

SCOP ID	family name	# proteins	# fprints	Remarks
46626	Cytochrome C	34	5	contain His-Cys-Cys triad; used in Figure 2
88854	Protein Kinase catalytic	29	30	small fingerprints
50847	Fatty Acid Binding (FABP)	11	32	used in Table 2

query list	short description	# proteins	Remarks
CASP5 T0137	17 homology models submitted for target T0137 in CASP5	17	All models on FABP template scored high (Table 2)
CulledPDB /background	CulledPDB at 90% sequence ID, 3 Å resolution, R-factor 1.0	6500	background dataset, also for non-redundant family members
sample background	Randomly selected sample of background proteins	620	Used to benchmark graph indexing

We report the match efficiency and hit rate for the CASP target dataset matched against the Fatty Acid Binding Protein fingerprint set in Table 2. Clearly, using the graph index improves the efficiency and hit rate. The improvement in the running time seems higher in cases where more subgraphs are found. To investigate this, we selected as a benchmark query about a tenth of our non-redundant background dataset, i.e. the first 620 proteins with between 42–1017 residues and between 117–5463 bonds in their AD(0.1) graph representations.

In Figure 2, we show the times taken to search for the 5 Cytochrome C fingerprints within the sample background dataset, with and without the graph index. Family fingerprints by definition are usually not found in the background, and the cumulative running times do not differ substantially in this case. We also picked 11 subgraphs with between 3 and 5 nodes that are frequent in the background – these 11 subgraphs have on the average 60 occurrences in our background dataset. Now the search using the graph index is still linear and takes about 1 second per query (0.014 second per occurrence found), while without the indexing only three subgraphs had been searched within one query in 30 minutes and the fourth did not finish running within 3 hours. This shows the scalability of our graph index to search many large dense proteins with multiple occurrences of subgraphs.

Query protein name	# fam FP	No graph index			Using local nhd graph index		
		Time (sec)	η	Hit rate	Time (sec)	η	Hit rate
AL025_1	17	107.14	0.02	0.59	1.52	0.23	0.85
AL044_1	17	86.91	0.03	0.59	1.12	0.27	0.89
AL397_1	17	110.15	0.02	0.59	1.90	0.24	0.85
AL400_1	2	65.12	0.04	0.09	0.64	0.07	0.33
TS011_1	3	133.15	0.03	0.13	0.32	0.11	0.27
TS070_5	6	69.01	0.03	0.26	0.22	0.21	0.67
TS086_1	16	107.27	0.02	0.55	0.83	0.26	0.80
TS086_4_1	1	2.59	0.05	0.05	0.06	0.12	0.25
TS132_1	4	109.68	0.03	0.17	0.61	0.21	0.44
TS139_1	4	118.54	0.01	0.17	0.25	0.17	0.36
TS203_1	17	111.51	0.02	0.59	1.91	0.24	0.85
TS231_1	2	4.50	0.03	0.09	0.07	0.15	0.40
TS231_4	1	41.19	0.09	0.05	0.14	0.15	0.50
TS233_1	17	99.85	0.07	0.59	1.98	0.23	0.85
TS233_2	2	10.49	0.08	0.13	0.37	0.05	0.50
TS282_2	4	87.69	0.03	0.17	0.92	0.10	0.50
TS283_1	6	48.85	0.03	0.26	0.28	0.20	0.67

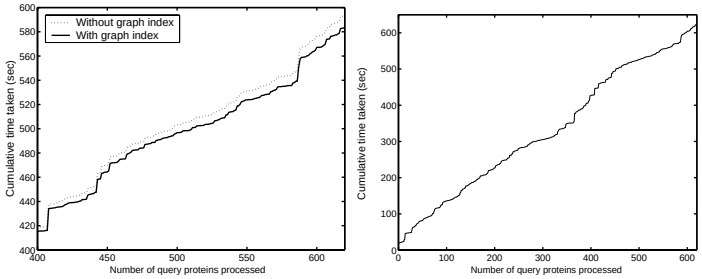


Figure 2. An easy case (left): searching for 5 fingerprints of the Cytochrome C SCOP family (all containing the functional His-Cys-Cys triad and additional neighboring residues) in 620 random proteins, a tenth of our CulledPDB dataset. Cumulative running time is plotted against cumulative number of query proteins. Though there were almost no matches, the graph index was still a little faster than no indexing. A hard/dense case (right): searching for 11 wholly hydrophobic subgraphs with 3–5 nodes that are frequent in 620 random proteins, a sample of our CulledPDB dataset. Each protein has on the average 60 occurrences of these 11 subgraphs. Comparing the two cases, we see that the graph index takes not much more time for this densely occurring set of subgraphs than for the sparsely occurring set. Without the graph index, however, this case is intractable; only 3 subgraphs could be searched in an 856-residue query within three hours of processing.

3.2. *Validating the Fingerprints by Cross Validation*

We validated the fingerprints of a family by cross-validation, i.e. by removing a subset of members of a family, finding the fingerprints from the rest and then an-

notating the removed members using the fingerprints derived from the remaining members. For example, in the serine proteases family dataset with 65 members, when we removed around 15% of the family (10 members), we got 907 fingerprints from the remaining 55 members. The 10 removed members have on average 856 of the 907 (94%) fingerprints. The ratio between the number of recovered fingerprints to the total number of fingerprints is defined as the *hit ratio*. We performed a standard 6 fold cross validation and the average hit ratio is 0.81 with standard deviation 0.07. This suggests that on average a true positive protein (the one that we know belongs to the family) is expected to contain around 80% of the total fingerprints we have. Given the fact that we usually have tens (more likely hundreds) fingerprints, we would expect that on average the any true positive proteins should contains enough fingerprints to be annotated correctly.

3.3. *Comparing Datasets from GO and SCOP classification*

We collected two groups of serine proteases from two independent sources. The first dataset we collected is from the Gene Ontology node: Trypsin activity molecular function (ID:0004295). We also obtained a group of serine proteases from two SCOP families: Eukaryotic Serine proteases (ID:40595) and Prokaryotic Serine Proteases (ID:50495) and mixed them together. We refer the first group as the GO SP dataset and the second one as the SCOP SP dataset. For both datasets, we used the cullpdb list to obtain the non-redundant structures. The two datasets are summarized in Table 3. It would be interesting to check whether these two different annotation systems agree with each other or not. For that purpose, we annotate the proteins from GO SP according to SCOP and the results are listed below. From the table, we found that the GO annotation and SCOP annotation agree with each other quite well.

3.4. *GO Enrichment Analysis of Protein Groups Featuring Fingerprints*

To further comparing the two classification systems, we carry the following experiment. In this experiment, given a list of proteins sharing a fingerprint from SCOP serine protease family, we try to evaluate how the functions of these proteins are distributed in GO. 72 fingerprints are discovered based on proteins from Serine Proteases Family. For each fingerprint, the proteins in the background dataset in which it occurs (*hits*) are extracted into a list and evaluated for GO enrichment. The size of the protein lists ranges from 60 to 97. We observe that the GO categories related to peptidase activity (shown in 3) are consistently enriched in all 72 protein lists, with p-value smaller than $1.0 * 10E - 15$.

The result suggests that proteins sharing the same fingerprint may have similar

SCOP	PDBID+Chain
UA	1md8a 1nl1a 1os8a 1p3ca 1p57a 1p57b 1pq7a 1q0pa 1qy6a 1s83a 1ssxa
49855	1nt0a 1nzia
50495	1agja 1arb0 1hpga 1l1ja 1qtfa 1sgpe 2sfa0 2sga0 **1ky9a 1l1ja 1lcy a**
50514	1ao5a 1autc 1azza 1bio0 1bqya 1brup 1cgha 1ddja 1dlea 1eaxa 1ekbb 1elt0 1elva 1eq9a 1eufa 1f7za 1fi8a 1fiwa 1fiza 1fizl 1fona 1fuja 1fxya 1g2la 1gg6b 1gg6c 1gj7a 1gj7b 1gvkb 1gvza 1h4wa 1h8dh 1h8dl 1hj8a 1hj9a 1iaua 1kdqa 1kdqb 1kigh 1klih 1lo6a 1ltoa 1m9ua 1mzaa 1nn6a 1npma 1orfa 1pfxc 1ppfe 1rfna 1rtfb 1ton0 1trna 1ucyj 1ucyk 2hlca 2pkaa 2pkab 3rp2a **1pytd**
54807	3proc
57197	1autl 1edmb 1g2lb 1kigl 1klil 1rfnb
57415	1bhta
57441	1i71a 1ki0a 1krn0 1pmla 3kiv0 5hpga
57631	1iodg 1j34c 1lqvc
74933	1ky9a 1lcya

```
GO:0008233 : peptidase activity (k:55/f:377)
        GO:0004175 : endopeptidase activity (k:55/f:297)
                GO:0004252 : serine-type endopeptidase activity (k:55/f:130)
                        GO:0004263 : chymotrypsin activity (k:52/f:73)
                        GO:0004295 : trypsin activity (k:55/f:85)
        GO:0008236 : serine-type peptidase activity (k:55/f142)
                GO:0004252 : serine-type endopeptidase activity (k:55/f:130)
```

Figure 3. Significantly enriched GO categories for a list of 62 proteins. Given a GO category, k is the number of proteins in the 62 proteins belonging to the category, and f is the number of proteins in the CulledPDB dataset belonging to the category.

molecular functions. Based on this observation, we can define a *functional neighbor* relation between families that share fingerprints, with strength proportional to the number of fingerprints shared. This method can be used to classify families of proteins with unknown function, or to derive new function annotations based on functional neighbors.

4. Conclusion

We developed a fast subgraph matching algorithm to match a protein family specific substructure (a fingerprint) to a large set of proteins structures. We evaluated the biological significance of such search using training sets from SCOP and Gene Ontology. Our results demonstrate that the fingerprints identified by our algorithm is stable and can be used to infer functions from unknown proteins.

References

1. V. Alexandrov and M. Gerstein. Using 3d hidden markov models that explicitly represent spatial coordinates to model and compare protein structures. *BMC Bioinformatics*, 5(1):2, 2004 Jan 9.
2. P. J. Artymiuk, A. R. Poirrette, H. M. Grindley, D. W. Rice, and P. Willett. A graph-theoretic approach to the identification of three-dimensional patterns of amino acid side-chains in protein structures. *Journal of Molecular Biology*, 243:327–44, 94.
3. J. Barker and J. Thornton. An algorithm for constraint-based structural template matching: application to 3d templates with statistical analysis. *Bioinformatics*, 19(13):1644–9, 2003 Sep 1.
4. A. Bateman, L. Coin, R. Durbin, R. D. Finn, V. Hollich, S. Griffiths-Jones, A. Khanna, M. Marshall, S. Moxon, E. L. L. Sonnhammer, D. J. Studholme, C. Yeats, and S. R. Eddy. The Pfam protein families database. *Nucl. Acids. Res.*, 32(90001):D138–141, 2004.
5. G. O. Consortium. The Gene Ontology (GO) database and informatics resource. *Nucl. Acids. Res.*, 32(90001):D258–261, 2004.
6. G. Dodson and A. Wlodawer. Catalytic triads and their relatives. *Trends Biochem Sci.*, 23(9):347–352, Sept. 1998.
7. M. A. Dwyer, L. L. Looger, and H. W. Hellinga. Computational Design of a Biologically Active Enzyme. *Science*, 304(5679):1967–1971, 2004.
8. J. S. Fetrow and J. Skolnick. Method for prediction of protein function from sequence using the sequence-to-structure-to-function paradigm with application to glutaredoxins/thioredoxins and t1 ribonucleases. *J. of Mol. Biol.*, 281:949–968, 1998.
9. J. Gibrat, T. Madej, and S. Bryant. Surprising similarities in structure comparison. *Curr Opin Struct Biol*, 6(3):377–85, 1996 Jun.
10. D. Haft, B. Loftus, D. Richardson, F. Yang, J. Eisen, I. Paulsen, and O. White. Tigrfams: a protein family resource for the functional identification of proteins. *Nucleic Acids Res*, 29(1):41–3, 2001 Jan 1.
11. L. Holm and C. Sander. Mapping the protein universe. *Science*, 273:595–602., 1996.
12. J. Huan, D. Bandyopadhyay, W. Wang, J. Snoeyink, J. Prins, and A. Tropsha. Comparing graph representations of protein structure for mining family-specific residue-based packing motifs. Journal of Computational Biology, 2004.
13. J. Huan, W. Wang, D. Bandyopadhyay, J. Snoeyink, J. Prins, and A. Tropsha. Mining protein family specific residue packing patterns from protein structure graphs. In *Eighth Annual International Conference on Research in Computational Molecular Biology (RECOMB)*, pages 308–315, 2004.
14. J. Huan, W. Wang, and J. Prins. Efficient mining of frequent subgraph in the presence of isomorphism. *ICDM*, 2003.
15. J. Huan, W. Wang, J. Prins, and J. Yang. Spin: Mining maximal frequent subgraphs from graph databases. *SIGKDD*, 2004.
16. S. Jones and J. M. Thornton. Searching for functional sites in protein structures. *Current Opinion in Chemical Biology*, 8:3–7, 2004.
17. R. A. Laskowski, J. D. Watson, and J. M. Thornton. From protein structure to biochemical function? *Journal of Structural and Functional Genomics*, 4:167–177, 2003.
18. I. Letunic, L. Goodstadt, N. Dickens, T. Doerks, J. Schultz, R. Mott, F. Ciccarelli,

R. Copley, C. Ponting, and P. Bork. Recent improvements to the smart domain-based sequence annotation resource. *Nucleic Acids Res*, 30(1):242–4, 2002.
19. T. Madej, J. Gibrat, and S. Bryant. Threading a database of protein cores. *Proteins*, 23(3):356–69, 1995 Nov.
20. M. Milik, S. Szalma, and K. Olszewski. Common structural cliques: a tool for protein structure and function analysis. *Protein Eng.*, 16(8):543–52., 2003.
21. A. Murzin, S. Brenner, T. Hubbard, and C. Chothia. SCOP: a structural classification of proteins database for the investigation of sequences and structures. *Journal of Molecular Biology*, 247:536–40, 1995.
22. R. Nussinov and H. J. Wolfson. efficient detection of three-dimensional structural motifs in biological macromolecules by computer vision techniques. *PNAS*, 88:10495–99, 1991.
23. C. Orengo, A. Michie, S. Jones, D. Jones, M. Swindells, and J. Thornton. CATH - a hierarchic classification of protein domain structures. *Structure*, 5(8):1093–1108, 1997.
24. R. B. Russell. Detection of protein three-dimensional side-chain patterns: new examples of convergent evolution. *Journal of Molecular Biology*, 279:1211–1227, 1998.
25. R. Samudrala and J. Moult. A graph-theoretic algorithm for comparative modeling of protein structure. *J Mol Biol.*, 279(1):287–302, 1998.
26. J. Skolnick, J. S. Fetrow, and A. Kolinski. Structural genomics and its importance for gene function analysis. *nature biotechnology*, 18:283–287, 2000.
27. A. Stark, S. Sunyaev, and R. B. Russell. A model for statistical significance of local similarities in structure. *Journal of Molecular Biology*, 326:1307–1316, 1998.
28. W. Taylor and C. Orengo. Protein structure alignment. *Journal of Molecular Biology*, 208:1–22, 1989.
29. M. Turcotte, S. Muggleton, and M. Sternberg. Automated discovery of structural signatures of protein fold and function. *J Mol Biol.*, 306(3):591–605.
30. J. R. Ullman. An algorithm for subgraph isomorphism. *Journal of the Association for Computing Machinery*, 23:31–42, 1976.
31. A. Via, F. Ferre, B. Brannetti, A. Valencia, and M. Helmer-Citterich. Three-dimensional view of the surface motif associated with the p-loop structure: cis and trans cases of convergent evolution. *Journal of Molecular Biology*, 303(4):455–465, Nov. 2000.
32. A. Wallace, N. Borkakoti, and J. Thornton. Tess: a geometric hashing algorithm for deriving 3d coordinate templates for searching structural databases. application to enzyme active sites. *Protein Sci*, 6(11):2308–23, 1997 Nov.
33. G. Wang and R. L. D. Jr. Pisces: a protein sequence culling server. *Bioinformatics*, 19:1589-1591, 2003. http://www.fccc.edu/research/labs/dunbrack/pisces/culledpdb.html.
34. P. Wangikar, A. Tendulkar, S. Ramya, D. Mali, and S. Sarawagi. Functional sites in protein families uncovered via an objective and automated graph theoretic approach. *J Mol Biol*, 326(3):955–78, 2003.
35. A. S. Yang and B. Honig. An integrated approach to the analysis and modeling of protein sequences and structures. ii. on the relationship between sequence and structural similarity for proteins that are not obviously related in sequence. *Journal of Molecular Biology*, 301(3):679–690, 2000.