FAST STRUCTURAL SIMILARITY SEARCH BASED ON TOPOLOGY STRING MATCHING*

SUNG-HEE PARK AND DAVID GILBERT

Bioinformatics Research Centre, Department of Computing Science, University of Glasgow, Glasgow, Scotland, G12 8QQ, UK.

KEUN HO RYU

Chungbuk National University, Cheongju, 361-763, R..O.Korea.

We describe an abstract data model of protein structures by representing the geometry of proteins using spatial data types and present a framework for fast structural similarity search based on the matching of topology strings using bipartite graph matching. The system has been implemented on top of the Oracle 9i spatial database management system. The performance evaluation was conducted on 36 proteins from the Chew and Kedem data set and also on a subset of the PDB40. Our method performs well in terms of the quality of matching whilst having the advantage of fast execution and being able to compute similarity search in polynomial time. Thus, this work shows that the pre-computed string representation of topological properties between secondary structure elements using spatial relationships of spatial database management system is practical for fast structural similarity search.

1 Introduction

The complexity of structure similarity search for 3D structures is NP hard [1]. Due to the complexity of similarity search and the exponential growth of the size of the 3D structure databases, the issue of speed becomes ever more important, and tools for fast structure similarity search have been required.

GRATH [2] and SSM [3] are examples of recent work dealing with the speed of structure similarity search which make full database searches possible. Some similarity search methods use fast pre-filtering before performing accurate alignment. Even in this case, due to the lack of pre-computed and stored data, each structure in the database has to be scanned at least once. Therefore, such an exhaustive search is expensive to perform.

The degree of abstraction influences the accuracy and performance of structure comparison programs. For example, DALI [4] and SSAP [5] which are based on atomic coordinate level data can be accurate but execute more slowly than less accurate approaches that use abstractions such as vectors and SSEs (e.g. VAST [6], LOCK [7], TOPS [8]).

We determined that the computation cost for geometrical features is known to be more expensive than that of I/O access to them whilst the opposite is true for general data. The computational cost of a topological match is less expensive than that of atomic-

* This work was supported by the Korean Research Foundation Grant funded by the Korean Government (MOEHRD). (KRF-2005-214-E00050).

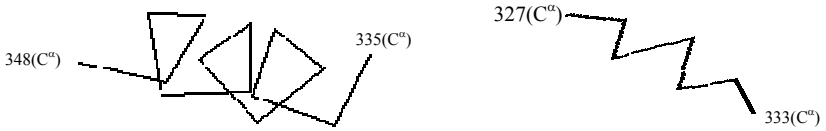
coordinate based geometry matching, and the comparison of topological properties can identify global similarities. Our method is based on the spatial theory of GIS (Geographic Information Systems): topological properties of spatial objects are invariant even if geometric features such as length and angle are easily changed [9]. Topological properties are preserved in conserved structures and can help to identify the fold family in similarity searching. Thus, we consider that topological properties of proteins are suitable for fast structure comparison.

In this paper we describe in detail a method to derive abstract models of protein structures at the tertiary (fold) level, where secondary structure elements (SSEs) are represented using spatial data types. Our models comprise sequences of topology strings, each string describing the topological relationship between a pair of such SSEs using the 9IM (Intersection Matrix) [10]. As an application of our abstract model to protein structure comparison, we describe a framework for fast similarity search based on finding maximum matching pairs of topology strings using a bipartite graph matching algorithm implemented by join operations. We report experimental results on two significant data sets, comparing our approach with the TOPS system.

2 An Abstract Model of Protein Structures

2.1. Geometry Representation of Proteins

The geometry of SSEs is modelled as a polyline of 4–10 connected points (the normal range of the numbers of amino acids in SSEs). These polylines are implemented using the geometric types of SDO_GEOMETRY in ORACLE 9i Spatial by LINE STRING where points are connected by straight line segments. Figure 1 (a) shows an example of Helix 4 that starts from amino acid residue 348 and ends at residue 335; Figure 1 (b) is for a strand 2 of protein domain 2bopA0. The tertiary structure of a protein can be defined as a finite set of SSEs and the spatial topological relationships between them, and can thus be described by a list of polylines.



(a) Helix 4 (348-335) (b) Strand 2 (327-333)

Figure 1. Polylines for an Helix and a Strand of 2bopA0

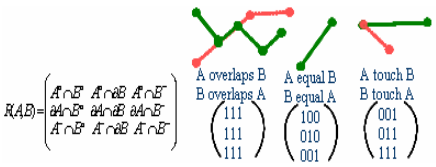
2.2. Inference of Topological Relationships

The biological meaning of protein topology refers to the spatial relations of 3D folds, i.e. the spatial arrangements of SSEs. This can be described using the topological relationships of spatial geometry and spatial types. Thus, the problem of inferring the topological relationships between SSEs in 3D space can be transformed into that of

inferring the spatial relationships between binary polylines in a spatial database. We employ spatial topological relationships [10] that have been previously used in GIS applications to infer relationships in 3D space. Spatial relationships between pairs of SSEs are represented by eight topological relations, defined by the 9IM (Intersection Matrix) [10].

Two lines can induce eight different types of topological relationships: *disjoint*, *contains*, *inside*, *equal*, *meet*, *cover*, *coveredBy*, and *overlaps*. In our approach we consider the three topological relations *touch*, *equal*, *overlaps* due to the fact that *overlaps* subsumes *inside, contains*, *cover, coveredBy*. These three topological relationships are enhanced by the addition of the types of SSEs, e.g. Helix and Strand. We thus generate the major nine topological relationships (e.g. Helix$\odot_{overlaps}$Helix, Helix $\odot_{overlaps}$Strand, Strand $\odot_{overlaps}$Strand, and etc.) as shown in Table 1. The three topological relationships (*overlaps, equal, touch*) are computed by a method that we have previously described [11], which uses topological operators in the ORACLE 9*i* Spatial. This system provides the spatial operator SDO_RELATE that implements a 9IM model between points, lines and polygons. The spatial operator is accomplished by the join operation of two R-tree indexes in a spatial query.

In the 9IM Model a binary topological relationship R(A, B) between two lines A and B is calculated by the comparison of A's interior (A°), boundary (∂A), and exterior (A–) with B's interior (B°), boundary (∂B), and exterior (B–). These six objects can be combined to form nine fundamental descriptions of the three topological relationships between two lines and can be concisely represented by the 3×3 matrix in Figure 2(a), called the 9-Intersection Matrix (9IM). The three topological relationships provide a mutually exclusive coverage with corresponding 9-intersection matrices in Figure 2(b) where 0 and 1 correspond to the empty set and non-empty set respectively.



(a) 9IM (b) three relationships

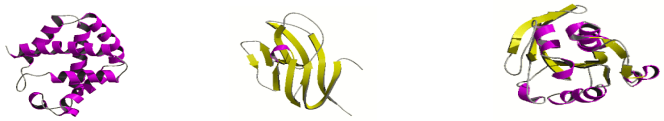
Figure 2. Topological Relationships

Table 1. Enhanced Topological Relationships

Topological Relation (⊙)	SSE types of Secondary Structure		
	Helix ⊙ Helix	Helix ⊙ Strand	Strand ⊙ Strand
overlaps	A	F	K
equal	B	E	H
touch	D	I	N

The topological relationship *overlaps* identifies a pair of SSE in a protein, where the interiors and boundaries of two SSEs intersect. The relationship *touch* recognizes that the boundaries intersect but interiors do not intersect between two SSEs whereas *equal* finds two SSEs having the same boundary and interior.

We formalize the binary topological relations corresponding to SSE types using these three topological relationships. Given a protein P={ S_1,…, S_i, …S_p}, where $i \in 1, 2,\ldots, p$, $0 < i < j < p$, p is the number of SSEs in a protein P, S_i and S_j are SSEs, the type



(a) 2hbg00 (b)1cdb00 (3)1aa900

(a) 2hbg00:
TOPS: NhHhhHhHC 1:4R4:6R5:7R
TStringSet: {(Anf,2,12,10.04)(Anf,2,14,8.1)(Asf,4,6,11.90)(Anf,4,8,9.117)(Anf,4,12,8.6)(Anf,8,12,14.3)(Anf,8, 14,15.2)(Anf,10,14,6.299)(Asf12.14.10.7)}
(b)1cdb00:
TOPS: NEEeeEC 1:5P2:3A2:4A4:5A
TStringSet: (Ksf,4,6,5.204)(Knf,4,8,3.601)(Ksf,8,10,5.202)}
(c)1aa900:
TOPS: NEheEhEhEhEhC 1:4P1:6P3:4A4:6R6:8Z8:10Z
TStringSet: {(Knf,2,8,3.92)(Knf,2,12,4.6)(Fsf,4,6,8.49)(Fnf,4,8,)(Anf,4,22,7.9)(Ksf,6,8,3.7)(Knf,12,16,3.92)(An f,14,18.5)(Knf,16,20,5.58)}

Figure 3. 3D Structures and Textural Representation

of an SSE is $T(S_i) \in \{\alpha, \beta\}$ standing for helix (α) and strand (β), and a set of topological relationships R={*overlaps*, *equal*, *touch*}, then:

[Definition1] (a binary topological relation) Let $R^2 = \{(S_i \odot S_j)\}$ be a binary topological relation, where $S_i, S_j \in P$, $i \neq j$, $\odot \in R$ and $T(S_i)$, $T(S_j) \in \{\alpha, \beta\}$. R^2_{ij} denotes a binary topological relationship $(S_i \odot S_j)$ between SSEs S_i and S_j.. The database terminology R^2 without subscripts is a binary topological relation over protein P, which is a set of binary topological relationships.

Most binary relationships are symmetric, but some are asymmetric. An asymmetric binary topological relationship, $R^2_{ij} = \{S_i \odot S_j\}$ is *forward* if $i < j$ and *backward* if $i > j$.

2.3. Notation of A Topology String and TStringSet

We represent our topological relationships by strings, called *topology strings* in order to reduce the complexity of the comparison problem, which is to find similar topological relationships between protein structures. A *topology string* is represented by a string of 3 characters, where:

- The alphabet of the first character denotes the type of topology relationship described in Table 1 (e.g. A, B, D, F, E, I, K, H, N).
- The second character encodes the order of the SSEs participating in a topology relationship. Sequential order, denoted by 's', means that SSE are adjacent along the backbone, whilst 'n' refers to neighboring order, i.e. one SSE is far from another in sequence (e.g. Asf or Anf).
- The third character denotes the direction of a topological relationship i.e. forward ('f') and backward ('b').

For instance, the string 'Anf' describes the Helix$\odot_{overlaps}$Helix topological relationship between two SSEs which are far from each other in the order of sequence, there being other SSEs in between the starting and ending helices in the relationship.

A protein structure is described by a (non-empty) set of topology strings, which we term a *TStringSet*. Each *topology string* together with its attributes are stored in a tuple. Our topology description can denote strand-strand and helix-helix relationships as well as helix-strand relationships that the current version of TOPS cannot represent.

2.4. Examples

Figure 3 illustrates examples of our representation of proteins and that of TOPS. SSE numbers and the distance between them are added to our topology strings. We observe that our string representation provides richer information about helix-helix interactions in protein 2hbg00 compared with TOPS.

3 Similarity Search

3.1. Topology String Match

A topology string match, i.e. matching a pair of SSE pairs, is the basic unit operation in this structure comparison. It evaluates the similarity between two topology strings.

[Definition 2] (a topology string match) The approximate string match function Match(a_i, b_j) returns string match score S_{ij} between given topology strings a_i and b_j..

Given a pair of topology strings (a_i, b_j), where $\{S_0, S_p\} \in a_i$, $\{S_q, S_r\} \in b_j$, and the topological relationships $R_{op} = (S_o \odot S_p)$ for the topology string a_i and $R_{qr} = (S_q \odot S_r)$ for b_j

Match(a_i , b_j) iff:

(i) $a_i = b_j$ if $T(S_o) = T(S_q)$, $T(S_p) = T(S_r)$ and $R_{op} = R_{qr}$

(ii) $0 \le |D(a_i) - D(b_j)| \le \sigma$

(iii) $0 \le |L(a_i) - L(b_j)| \le \varepsilon$

(iv) $Dir(a_i) = Dir(b_j)$ (1)

where,

σ: the tolerance for distance difference between the matched strings a_i, b_j ($0 < \sigma < 4$)

$D(a_i)$: distance between SSE S_0 and SSE S_p in a topology string a_i

$D(b_j)$: distance between SSE S_q and SSE S_r in a topology string b_j

ε: difference between the SSE length ratios for a pair of topology string a_i and b_j ($0 < \varepsilon < 0.5$)

$L(a_i)$: the length ratio of SSEs joined in a topology string a_i

$L(b_j)$: the length ratio of SSEs joined in a topology string b_j

$L(a_i) = MinLen(S_0, S_p)/MaxLen(S_0, S_p)$ and $L(b_j) = MinLen(S_q, S_r)/MaxLen(S_q, S_r)$

$MinLen(S_0, S_p)$: The smaller length of SSEs S_0 and S_p in a topology string a_i

$MaxLen(S_0, S_p)$: The greater length of SSEs S_0 and S_p in a topology string a_i

$MinLen(S_q, S_r)$: The smaller length of SSEs S_q and S_r in a topology string b_j

$MaxLen(S_q, S_r)$: The greater length of SSEs S_q and S_r in a topology string b_j

$Dir(a_i)$: a relative direction {P, A} of SSE S_0 and S_p in a topology string a_i

$Dir(b_j)$: a relative direction {P, A} of SSE S_q and S_r in a topology string b_j

The distance decides the degree of overlap between SSEs in a pair of topology strings. A low distance difference indicates a high degree of similarity between two pairs of SSEs.

In our experiments we used values of 2 for σ and 0.3 for ε, which are the default for these parameters when measuring the similarity between two proteins.

3.2. Pairwise Comparison

The similarity of a pair of protein structures in our representation is measured by the similarity between their corresponding *TStringSets*. A pairwise operation is employed to find the maximum subsets of matched pairs of topology strings between two sets of topology strings.

[Definition 3] (pairwise comparison: matching of two *TStringSets*) Given two sets of topology strings (*TStringSets*) for a query protein Q, Q= $\{a_1, \ldots, a_n\}$ and a target protein T in database D, T =$\{b_1, \ldots, b_m\}$, pairwise comparison S(Q, T) is an operation to find the maximum number of weighted matching pairs of topology strings S(Q, T) = $\{(a_1^1, b_1^1), (a_2^2, b_2^2), \ldots, (a_i^k, b_j^k)\}$, $1 \le k < n, m$ and returns the associated similarity score S.

The pairwise comparison problem is transformed into a maximum matching problem in a bipartite graph. Given the topology string sets Q and T, a directed weighted bipartite graph G = (V, E) can be constructed as follows: $V = V_Q \cup V_T$, $E = \{e_{ij}\}$. Each edge e_{ij} (i.e. $i = 1,2,\ldots, n$; $j = 1,2, \ldots, m$) corresponds to a weighted link between a_i and b_j, whose weight $w(e_{ij})$ is equivalent to the similarity S_{ij} between topology string a_i , b_j. A graph is bipartite if it has two kinds of vertices and edges are only permitted between vertices of different kinds. Thus the graph created is a weighted bipartite graph.

Our similarity score function for pairwise comparison is based on a compression measure over topology strings. It is calculated by the sum of topology string matching scores as follows:

[Definition 4] (Similarity score) The similarity score *S* for two sets of the topology strings for a query protein Q=$\{a_1, a_2, \ldots, a_i, \ldots, a_n\}$ and a target protein T=$\{b_1, b_2, \ldots, b_j, \ldots, b_m\}$, is computed by Eq. (2).

$$S = \sum_{k=1}^{n} \sum_{l=1}^{m} \left(\frac{S_{kl}}{Max\ (|\ Q\ |, |\ T\ |)} \right) \cdot (\log_e |\ M^{QT}\ |) \qquad (2)$$

iff Match(k, l) and $|M^{QT}| \ge 2$

where,

k: a topology string in the query protein Q, i.e. $k \in \{a_1, a_2, \ldots, a_i, \ldots, a_n\}$

l: a topology string in the target protein T, i.e. $l \in \{b_1, b_2, \ldots, b_j, \ldots, b_m\}$

$|Q|$: total number of topology strings in a query protein Q

$|T|$: total number of topology strings in a target protein T in a database

$|M^{QT}|$: number of the matched topology strings between proteins Q and T

S_{kl} : The topology string match score S_{kl} for topology strings $k \in$ Q and $l \in$ T

High structural similarity scores do not always imply high significance of an alignment. Thus, we use the log ratio of the number of the matched strings in order to assign less weight for low numbers of matched topology strings. The best known strongly polynomial time bound algorithm for weighted bipartite matching is the classical

Hungarian method due to Kuhn [12], which runs in time $O(|V|(|E| + |V|)\log|V|)$. Weighted bipartite matching algorithms can be implemented efficiently, and can be applied to graphs of reasonably large size. The pairwise comparison can run in time $O(n(m + n\log n))$, where a bipartite matching graph $G = (V_Q \cup V_T, E)$, $n = |V_Q \cup V_T|$, $m = |E|$, n is number of vertex in query protein Q and a protein T in a database, m is the number of edges and the maximum edge capacity is 1. Our pairwise comparison algorithm can be solved in polynomial time.

4 Experiments

4.1. Datasets

In order to permit a large scale analysis, a protein 3D structure database was constructed from a subset of domains in the SCOP PDB40 [13] release 1.61 representative dataset. This release of the PDB40 contains proteins no two of which share >40 % sequence similarity, a total of 4,420 domains in the all-α, all-β, α/β, and α+β SCOP classes. We selected the subset of entries for which we had sufficient data to construct our models, a total of 2,654 domains.

Table 2. Data distribution in PDB 40

SCOP Class	Fold Family	DOM	HDP	% of HDP	NHDP	% of NHDP
All α	171	415	1,742	2.12	82,238	97.88
All β	124	457	5,074	5.35	94,810	94.65
α/β	172	904	10,104	2.56	395,008	97.44
α+β	284	878	2,609	0.69	376,983	99.31
Total	751	2,654	19,529	2.02	949,039	97.98

DOM: Number of unique protein domain entries
HDP: Number of homologous domain pairs
NHDP: Number of non-homologous domain pairs

Table 3. Fold Superfamily in Chew and Kedem

Supferfamily	Fold Class	Num. of DOM
1.1.1	Globin Like	16
1.107.1	All a	1
1.34.1	All a	1
2.1.1	All b	7
3.1.11	TIM barrels	3
3.1.15	TIM barrels	1
3.25.1	a/b	5
3.26.2	a/b	1

Our data set selected from SCOP PDB40 contains 75l SCOP fold superfamilies (Table 2); the distribution of SCOP fold superfamilies in our data set is diverse. We generated a set of all against all domain pairs for our 2,654 unique domain entries and selected a subset for evaluation of ~1M pairs comprising those pairs for which both domains belong to the same SCOP class. We also tested our method against the data set of Chew and Kedem [14] for a smaller scale evaluation of its accuracy. Table 3 shows 35 domain entries in eight different SCOP superfamilies of the three SCOP classes. Experimental data sets are available at http://www.brc.dcs.gla.ac.uk/~shpark/pcom/.

4.2. Evaluation

We compared our results with the SCOP classification hierarchy. We have constructed ROC curves and calculated the corresponding AUC (Area Under ROC Curves) values over different SCOP classes (Figure 4) for both data sets. To construct the ROC curve, we define that two domains are defined as homologous if at least their first three SCOP

numbers are identical; the domains are non-homologous if only their first SCOP numbers are identical. We did not include protein domain pairs for which only the first two levels of their SCOP numbers match since the SCOP classification does not differentiate between homologous and non-homologous pairs at this fold level.

Chew & Kedem data sets: We evaluated the accuracy of our method by performing a one against all comparison for each of the 35 domains; the average AUC value for this is 87.93%. Figure 4(a-c) gives ROC curves for three illustrative one against all domain comparisons. Table 4 shows the results for our method regarding the coverage of the correct fold superfamily at the top of our resulting lists sorted by comparison score for those superfamilies that had more than one domain member. We repeated the same experiment for TOPS (Table 5). Overall, our system consistently finds the correct fold superfamily within the top 3 positions in the list with 85.7 % of certainty. Regarding the test for major SCOP fold classes, our algorithm performs better than TOPS for all-α proteins and the α/β Rossman folds (family 3.25.1). TOPS performs better for the all-β class and for the 3.1.11 α/β fold family (TIM barrels).

Table 4. Coverage of top K in our result lists

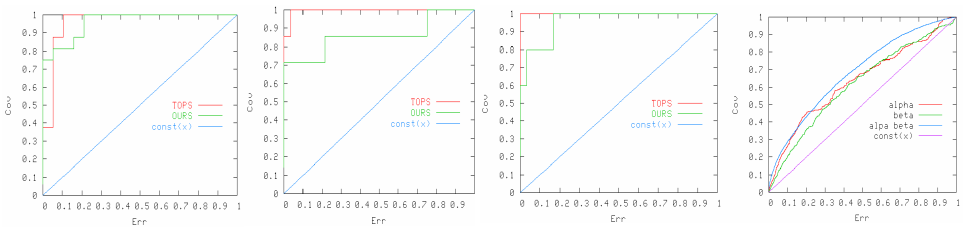
fold family	AUC value[a] (%)	Top K[b]=3		Top K[b] =4		Top K[b] =5	
		TP[c]	FP[d]	TP[c]	FP[d]	TP[c]	FP[d]
1.1.1	88.1	2.9	0.1	3.82	0.18	4.8	0.2
2.1.1	84.6	2.4	0.57	3	1	3.6	1.4
3.1.11	85.4	1.6	1.3	2	1	2	1
3.25.1	84.7	2.8	0.2	3.8	0.2	3.8	0.2

Table 5. Coverage of top K in TOPS

Fold family	AUC value[a] (%)	Top K[b]=3		Top K[b] =4		Top K[b] =5	
		TP[c]	FP[d]	TP[c]	FP[d]	TP[c]	FP[d]
1.1.1	82.15	2.9	0.1	3.8	0.2	4.8	0.2
2.1.1	97.85	3	0	4	0	5	0
3.1.11	94.58	2	1	2.3	0.7	2.6	0.4
3.25.1	81.9	2.6	0.4	3.2	0.8	3.2	0.8

[a]: Average AUC value in a corresponding family for one against all search
[b]: Number of top K domain in result lists of similarity score.
[c]: Number of average times ranks true positives that is homologous pairs ahead of non-homolog pairs
[d]: Number of average times ranks false positives

Our explanation is that since our topology strings encode geometrical features such as the distance between SSEs, and helix-helix and helix-strand interactions, our comparison performed well for the all-α class in the experiments. However, TOPS strings mainly encode β-strand connectivity [8]. Therefore TOPS works well for superfamilies in the all-β class as well as for those in the α/β class whose structures are classified according to β sheet topology (e.g. TIM barrels).



(a) 2hbg00((1.1.1) (b) 1cdb00 (2.1.1) (c) 1aa900 (3.25.1) (d) SCOP PDB40

Figure 4 . ROC Curves

Subset of SCOP PDB40: The AUC values for α, β, and mixed α β classes are 68.5%, 62.6%, and 69.4% respectively, with an overall average of 66.83%. The corresponding ROC curves are shown in Figure 4(d). The reason that the accuracy is lower than for the Chew & Kedem data set is related to the distribution of the SCOP PDB40 data, which is extremely diverse and contains a very low number of homologous pairs.

4.3. Performance Analysis

Our similarity search algorithms ran on an IBM PC with 3GHz CPU and 2GBytes memory. We used the database mentioned above as our target database which was built on ORACLE 9*i* DBMS by using spatial types and topological operators running on a spatial index.

The ~1M pairwise comparisons took 10hr and 17 min and all against all for 35 domain entries took 8.1 sec. The average execution time for pairwise comparisons ranged from 12.17 ms in the small scale analysis to 38.45 ms in the large scale analysis. Although our method has a very fast comparison time, it took 1 hour and 20 minutes to represent the geometry of SSEs using spatial types and build an index over 2,645 protein domain having 34, 114 SSEs (Table 6). According to our experiments, it takes on average 1.08 sec to build the topology strings for one protein. Most of this time is in the discovery of binary topology strings.

Table 6. Execution time for computational task

Task name	Num. of Data	Total exe.time	AVG exe. Time
SSE representation	2,654 dom. ent.	1hr 20 m	1.8 sec
Generation of topology string	2,654 dom. ent.	1hr 05m	1.5 sec
All against all search	35 dom ent.	8.1 sec	12.17ms
pairwise comparison	968,568 dom. pairs	10hr17m	38.45ms

As shown in Table 6, the computation time for the identification of topological properties based on geometrical features is expensive to run. However, since the database of topology strings is pre-computed, the time required for the discovery of topology strings does not affect the execution time for structural similarity search. The matching of two topology string sets is performed in polynomial time.

5 Conclusions

We have developed a new fast similarity search based on topology string matching. We used a constrained string match algorithm for the comparison of linear topology strings encoding the non-linear topological relationships. Our method could be used as a filtering step prior to slower but more accurate similarity search and fold discovery methods based on all-atom approaches in a pipeline for automated structure classification.

Our results indicate that our algorithm performs reasonably well for structures with all α-helices and a mixture of α and β SSEs, but does not work so well for all-β proteins. The overall accuracy of our system is comparable to that of TOPS, which is a fast and

accurate structure comparison method based on abstraction over SSE topologies. The weakness of TOPS is that it does not perform well on α class members; our method is superior in such cases.

Compared with existing methods, our method uses spatial characteristics of protein structures that are represented by using an existing spatial DBMS. The application of spatial databases in bioinformatics has many advantages; one can analyze and represent multi-dimensional information with spatial types, multidimensional index, and spatial operations that are implemented using algorithms from computational geometry.

We observe that the use of index structures will facilitate the retrieval of biological data, and extracted patterns from source databases. In future research, we will work on the development of index based similarity search over topology strings using spatial indexes.

Acknowledgements

We thank Mallika Veeramalai of the Bioinformatics Research Centre at the University of Glasgow and Juris Viksna of the University of Latvia for helpful discussions about use of bipartite graph matching.

References

1. Goldman,D. (2000) Algorithmic Aspects of Protein Folding and Protein Structure Similarity, PhD Thesis, Department of Computer Sciences, UC Berkeley.
2. Harrison, A., Pearl, F., Sillitoe, I., Slidel, T., Mott, R., Thornton J. and Orengo, C. (2003) Recognizing the fold of a protein structure, J. of Bioi., 19, 748.
3. Krissinel. E and Henrick. K. (2004) Secondary structure matching, a new tool for fast protein structure alignment in three dimentions, Crystallographica Section D Biological Crystallography, 60, 2258.
4. Holm, L. and Sander, C. (1993) Proein structure comparison by alignment of distance matrices, J. of Mol. Biol., 233, 123.
5. Orengo, C. A. and Taylor, W.R. (1996) SSAP: sequential structure alignment program for protein structure comparison, J. of Meth. in Enzy., 266, 617-635.
6. Gibrat, J-F., Madej, T. and Bryant, H. (1996) Surprising similarities in structure comparison, Curr.Opin.inStru.Bio., 6, 377.
7. Singh, A.P.and Brutlag, D.L. (1997) Hierarchical protein structure superposition using both secondary structure and atomic representations. Proc. Int. Conf. Intell. Syst. Mol. Biol. 5, 284.
8. D. Gilbert, D. R. Westhead, J. Viksna, and J. M. Thornton. A computer system to perform structure comparison using TOPS representations of protein structure. J.of Comput. Chem., 26:23-30, 2001.
9. Laurini, R.and Thompson, D. (1992) Fundamentals of Spatial Information Systems, Academic Press Ltd.
10. Egenhofer, M., Frank, A. and Jackson, J. (1989) A topological data model for spatial databases. In Proceeding of the Symposium on the Design and Implementation of Large Spatial Databases '89, Springer-Verlag, LNCS 409, 271.
11. Park, S.H., Ryu, K. H. and Gilbert, D. (2005) Fast Similarity Search for 3D Protein Structures using Topological Pattern Matching based on Spatial Relations IJNS, 15(4), 287.

12. Kuhn, W., (1955) The hungarian method for the assignment problem, Naval Research Logistics Quarterly, 2, 83.
13. Murzin, A.G., Brenner, S.E., Hubbard, T. and Chothia, C. (1995) SCOP: a structural classification of proteins database for the investigation of sequences and structures. J. Mol. Bio., 247, 536.
14. Chew,L.P. and Kedem, K. (2004) Finding consensus shape for a protein family. Algorithmica, 38, 115-129.