BIOINFORMATICS ORIGINAL PAPER Vol. 24 no. 5 2008, pages 645–651
doi:10.1093/bioinformatics/btm641

Structural bioinformatics

Structural search and retrieval using a tableau representation of protein folding patterns

Arun S. Konagurthu[1,*], Peter J. Stuckey[2,3] and Arthur M. Lesk[1,*]

[1]Department of Biochemistry and Molecular Biology and The Huck Institute for Genomics, Proteomics and Bioinformatics, The Pennsylvania State University, University Park, PA 16802, USA, [2]Department of Computer Science and Software Engineering and [3]NICTA Victoria Laboratories, The University of Melbourne, Victoria 3010, Australia

Received on November 14, 2007; revised on December 14, 2007; accepted on December 29, 2007

Advance Access publication January 5, 2008

Associate Editor: Keith Crandall

ABSTRACT

Comparison and classification of folding patterns from a database of protein structures is crucial to understand the principles of protein architecture, evolution and function. Current search methods for proteins with similar folding patterns are slow and computationally intensive. The sharp growth in the number of known protein structures poses severe challenges for methods of structural comparison. There is a need for methods that can search the database of structures accurately and rapidly.

We provide several methods to search for similar folding patterns using a concise tableau representation of proteins that encodes the relative geometry of secondary structural elements. Our first approach allows the extraction of identical and very closely-related protein folding patterns in *constant-time* (per hit). Next, we address the hard computational problem of extraction of maximally-similar subtableaux, when comparing two tableaux. We solve the problem using Quadratic and Linear integer programming formulations and demonstrate their power to identify subtle structural similarities, especially when protein structures significantly diverge. Finally, we describe a rapid and accurate method for comparing a query structure against a database of protein domains, `TableauSearch`. `TableauSearch` is rapid enough to search the entire structural database in seconds on a standard desktop computer. Our analysis of `TableauSearch` on many queries shows that the method is very accurate in identifying similarities of folding patterns, even between distantly related proteins.

Availability: A web server implementing the `TableauSearch` is available from http://hollywood.bx.psu.edu/TabSearch

Contact: arun@bx.psu.edu, aml25@psu.edu

Supplementary information: Supplementary Data are available at *Bioinformatics* online.

1 INTRODUCTION

The growth of experimentally determined protein structures creates challenges and opportunities for organizing and classifying their folding patterns and for searching for similar structures. As of date (October 23, 2007), 46 679 proteins are recorded in the protein data bank (PDB) (Berman *et al.*, 2002). The latest release of ASTRAL SCOP database (v1.71) (Chandonia *et al.*, 2004) contains 75 632 domains classified into 971 folds. CATH (v3.1.0) (Orengo *et al.*, 1997), another fold classification database, contains 93 885 domains classified into 1084 different folds.

This rise in number of known structures makes comparison of structures demanding in time. Current methods for protein database search and classification are either based on structural alignment methods at the level of individual residues (Orengo *et al.*, 1997; Holm and Sander, 1993) or on the geometry of pairs of secondary structural elements (Abagyan and Maiorov, 1988; Artymiuk *et al.*, 1992a, b; Grindley *et al.*, 1993; Harrison *et al.*, 2003; Koch *et al.*, 1996; Lesk, 1995; Madej *et al.*, 1995; Mizuguchi and Go, 1995; Rufino and Blundell, 1994; Shi *et al.*, 2007). Structural alignments, which search for detailed residue–residue correspondences, are extremely slow to scour through the large database of protein structures (Holm and Sander, 1993; Konagurthu *et al.*, 2006; Orengo and Taylor, 1990; Shindyalov and Bourne, 1998).

Methods which use the geometric profiles of secondary structural interactions are generally faster than atom-level or even residue-level structural alignment methods. Similarities of folding patterns are visible even in coarse-grained structure representations using cartoons of secondary structural elements (helices and strands of sheets). This suggests that the essence of a folding pattern is captured in the geometry of interactions of pairs of secondary structural elements (SSEs). Most methods which explore the secondary structural geometry of protein folds depend mainly on graph–theoretic techniques such as maximal common subgraph isomorphisms or clique extraction, to identify similarities (Artymiuk, 1992a; Grindley *et al.*, 1993; Harrison *et al.*, 2003; Koch *et al.*, 1996; Rufino and Blundell, 1994), which are computationally hard (Papadimitriou and Steiglitz, 1998).

This article presents methods which rapidly search the entire protein structural database with high sensitivity, to be able to identify even distantly related protein folding patterns. The methods are based on a simple and robust encoding of the geometry of interactions of helices and strands of sheet in a square symmetric matrix or tableau (Lesk, 1995). Kamat and Lesk (2007) observed that tableaux robustly represent the

*To whom correspondence should be addressed.

folding patterns in ASTRAL. The tableau encoding discretizes the representation of any folding pattern and hence facilitates the use of pattern matching algorithms for fold identification and similarity detection.

In this article, we describe several methods for accurate and fast retrieval of similar protein folds. We generalize the definition of a tableau from Lesk (1995) to include the relative geometry of all pairs of secondary structural elements, not limited to the pairs of elements which are in contact. We introduce tableau hashing that allows retrieval of identical and near-identical structures in a constant time ($O(1)$) look-up (per hit). We then explore the computationally hard problem of extraction of maximally-similar subtableaux when comparing the tableaux of two proteins. We solve this problem using very efficient Quadratic and Linear integer programming formulations. This is required to identify similarities among divergent proteins, which have lower similarity even at the tableau level. Finally, we present a rapid yet sensitive database search method, `TableauSearch` which can scan the entire protein domain database in seconds with high accuracy.

We note that the identification of proteins with common folds, based on similarities of tableaux, does not automatically induce a residue–residue alignment. After extracting structurally-similar regions, we use MUSTANG for structural alignment (Konagurthu *et al.*, 2006).

2 MATERIALS AND METHODS

We calculated tableaux for all 75 632 domains in the ASTRAL SCOP 1.71 database (Chandonia *et al.*, 2004). ASTRAL inherits its definitions of domains from SCOP (Lo Conte *et al.*, 2000). `DSSP` (Kabsch and Sander, 1983) was used to define the secondary structure. `MD5sum` (Rivest, 1992) was used to produce the `MD5` hashes of the tableaux. All our software was written in `C++`. Quadratic and Linear integer programming formulations were solved using `ILOG CPLEX Concert Technology`[1] libraries for `C++`.

3 REPRESENTATION OF PROTEIN FOLDING PATTERNS AS TABLEAUX DISPLAYING GEOMETRIC RELATIONSHIPS OF HELICES AND STRANDS

In this section, we describe the details of construction of a tableau representation of the structural information of a protein.

3.1 Finding secondary structural elements

For a protein structure $\mathcal{P}$, let $\{e_1, \ldots, e_n\}$ be the set of n secondary structural elements (SSEs) in order of their appearance from the N- to C- terminus of $\mathcal{P}$. `DSSP` (Kabsch and Sander, 1983) categorizes SSEs as either helices (labeled here as α) or strands of a sheet (β). We do not differentiate among various types (α, 3_{10}, π) of helices.

3.2 Finding the inter-axial geometry of helices and strands

The relative orientation of two SSEs specifies an angle $-180^\circ \leq \omega \leq 180^\circ$ as the angle between the vectors along the axis of a helix, or along the least-squares line through the C_α atoms of a strand in the direction $\mathrm{N} \rightarrow \mathrm{C}$, projected on a plane normal to their mutual perpendicular (Fig. S1).

[1]http://www.ilog.com/products/cplex/

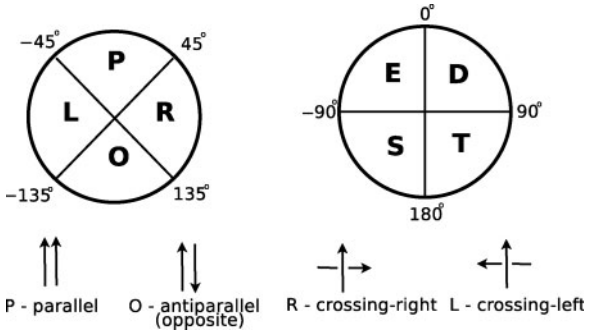


Fig. 1. Double-quadrant encoding of angles recorded in tableaux. Note that crossing-left and crossing-right are distinguished; tableaux do contain enantiomorph information.

By calculating the relative orientation of each pair of SSEs $\in \mathcal{P}$, we build an orientation matrix $\Omega = (\omega_{ij})_{1 \leq i,j \leq n}$.

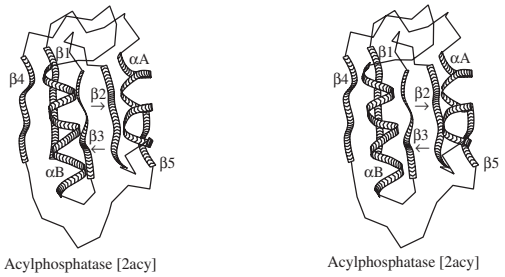
3.3 Tableau encoding of the orientation matrix

A tableau encodes the relative orientation of each pair of SSEs. Because proteins with similar folding patterns show considerable variability in the angles between pairs of SSEs (Lesk and Chothia, 1980), an encoding scheme should characterize the geometries by broad rather than narrow categories. Lesk (1995) proposed a double-quadrant encoding scheme (Fig. 1).

The use of ranges rather than continuous values of ω keeps the representation discrete. The possible range of any interaxial angle ($-180^\circ < \omega \leq 180^\circ$) is divided into quadrants in two different ways, differing in orientation by 45°. Any angle ω between two SSEs lies within a quadrant in each of these two partitions of the circle. The quadrants labelled P, O, R and L, in the division of the circle on the left of Figure 1, are centred around the relative orientations shown at the bottom of the Figure 1. E, D, T and S label the rotated set of quadrants (Lesk, 1995).

For adjacent strands of the same β sheet, additional two letter codes KK and HH specify parallel and anti-parallel β sheet interactions respectively. This is useful to distinguish strands that form a β sheet, from those in different β sheets packed face to face.

Classifying any relative orientation angle ω according to both divisions of the circle gives a discrete representation of relative orientation as a two-character code. For example, $\omega = 158^\circ$ corresponds to the encoding OT. If ω were represented by a single character, based on either partition of the angle-wheel, then two values of ω differing even infinitesimally might belong to different classes. Single-character encoding would make it difficult to identify similar geometric relationships by matching corresponding angles in tableaux. However, values of ω differing by $\leq 45^\circ$ cannot lie in different quadrants in both divisions of the circle, hence making it useful to spot similarities while taking into account the observation that relative orientations of secondary structure elements in homologous proteins can change substantially although the basic topology of folding pattern is retained (Lesk and Chothia, 1980).



(**a**)

β_1	156.5	−21.3	153.9	5.4	160.8	−34.5
156.5	α_A	172.9	175.9	156.1	−41.3	168.9
−21.3	172.9	β_2	−173.7	−20.1	139.7	−14.8
153.9	175.9	−173.7	β_3	154.2	−44.5	171.3
5.4	156.1	−20.1	154.2	α_B	−157.6	−34.4
160.8	−41.3	139.7	−44.5	−157.6	β_4	127.6
−34.5	168.9	−14.8	171.3	−34.4	127.6	β_5

(**b**) Orientation Matrix, Ω

β_1	OT	KK	HH	PD	HH	KK
OT	α_A	OT	OT	OT	PE	OT
KK	OT	β_2	HH	PE	HH	KK
HH	OT	HH	β_3	OT	KK	HH
PD	OT	PE	OT	α_B	OS	PE
HH	PE	HH	KK	OS	β_4	RT
KK	OT	KK	HH	PE	RT	β_5

(**c**) Tableau, T

Fig. 2. (**a**) Structure of acylphosphatase (in stereo), an $\alpha\beta$-protein (Protein Data Bank Code 2ACY). The chevrons indicate the direction of the polypeptide chain, from N- to C-terminus. (**b**) the matrix Ω containing the relative geometry of all pairs of SSEs, and (**c**) its corresponding tableau T. The labels of rows and columns are denoted in their main diagonals of the matrices.

Formally, let the tableau T be represented as $T=(t_{ij})_{1\leq i,j\leq n}$, where any t_{ij} is the two-character string corresponding to the two-quadrant encoding of the angle ω_{ij}. The diagonal elements t_{ii}, $\forall 1\leq i\leq n$ represent self-relationships, hence are instead used to record labels of the SSEs. The traditional convention labels successive helices with letters $(\alpha_A, \alpha_B, \ldots)$ and strands by numbers $(\beta_1, \beta_2, \ldots)$. Tableaux are symmetric matrices (Fig. 2).

4 CONSTANT-TIME STRUCTURAL LOOK-UP TECHNIQUES

Tableau-hashing methods allow the identification of identical and closely-related folding patterns in constant-time.

4.1 Checking for identical tableaux

From the tableaux corresponding to all the domains in the ASTRAL SCOP database, `MD5` (Message-Digest algorithm 5) (Rivest, 1992) hashes are computed. This provides, in effect, a 'hash code' for tableaux, from which a table of protein domains corresponding to each unique `MD5` hash is derived. Given a new structure, we compute its tableau. The `MD5` hash of the tableau corresponding to the new structure permits retrieval of structures with identical tableaux in constant time.

4.2 Extension of constant-time hash coding to non-identical tableaux

It is straightforward to extend this method to retrieve tableaux that differ in only one row and column. For example, at the secondary structure level the difference between Human α-haemoglobin and Human β-haemoglobin is the absence of the D-helix (α_D) in the α subunit. To retrieve such closely related folds: for each tableau, containing N rows and N columns, $N-1$ subtableaux are created by deleting, separately, each row and the corresponding column. The database of `MD5` hashes is augmented by adding the hashes of all these subtableaux (containing one row-column deletion). Looking up in this augmented database, the hashes corresponding to the tableau computed from a new protein structure (and all the subtableaux derived from that tableau by deleting, separately, each row and the corresponding column) will identify all known structures that either have identical tableaux to the novel protein, or which differ by the addition or subtraction of exactly one additional secondary structure element. This will multiply the size of the database by roughly an order of magnitude, which makes no great demands on resources. Since look-up requires time independent of the size of the database, no time penalty is incurred after the preprocessing is completed.

The database of hashes can be further augmented by deleting from each tableau, separately, all *pairs* of rows and the corresponding columns. In this way, we can in principle generate successions of neighbors of each tableau by deleting (almost) all possible subsets of rows (and corresponding columns). However, this would put a enormous load on the size of the database of tableau hashes. Moreover, ultimately the deletions would reduce the representation of protein folding patterns to individual supersecondary structures, similarity of which is an inadequate criterion for homology. For the work in this article we stop at deleting pairs of rows (and corresponding columns). Therefore, the constant-time structural look-up for the database we have prepared allows the identification of identical and similar tableaux and hence protein structures.

5 RIGOROUS METHODS FOR COMPARING TWO TABLEAUX

Next, consider the problem of extraction of *maximally-similar subtableaux* shared between any two tableaux. From the perspective of computational complexity, this problem is NP-hard as it is equivalent to the quadratic assignment problem. However, the dimensions of tableaux are very small. Hence, solutions of this problem for tableaux will be practical.

We present here two methods to extract such similarities. The first method uses a Quadratic integer programming (QIP) formulation of the problem. The second method formulates the same problem equivalently as an integer linear program (ILP). Both these equivalent formulations extract maximally-similar subtableaux.

Let $\mathcal{P}^q \equiv (e_1^q, \cdots, e_{N^q}^q)$ denote the query protein which contains N^q SSEs. Assume that $\mathcal{P}^q$ is being compared to another protein $\mathcal{P} \equiv (e_1, \ldots, e_N)$ in the database, containing N SSEs. Let $\{\Omega^q = (\omega_{ij}^q)_{1\leq i,j\leq N^q}, T^q = (t_{ij}^q)_{1\leq i,j\leq N^q}\}$ and $\{\Omega = (\omega_{ij})_{1\leq i,j\leq N}\}$, $T=(t_{ij})_{1\leq i,j\leq N}\}$ be the {orientation, tableau} matrices of $\mathcal{P}^q$ and $\mathcal{P}$, respectively.

5.1 Quadratic integer programming-based extraction of maximally-similar subtableau

We introduce Boolean variables y_{ij}, $1 \leq i \leq N^q$, $1 \leq j \leq N$, where $y_{ij}=1$ indicates that the ith SSE $\in \mathcal{P}^q$ is matched with jth SSE $\in \mathcal{P}$, and $y_{ij}=0$ indicates they are *not* matched.

The QIP formulation for comparing two tableaux for similarities is as follows:

$$\text{maximize} \quad f(y) = \sum_{1\leq i,k\leq N^q, 1\leq j,l\leq N} \zeta\left(t_{ik}^q, t_{jl}\right) y_{ij} y_{kl}, \qquad (1)$$

subject to

$$\sum_{j=1}^{N} y_{ij} \leq 1, \qquad 1 \leq i \leq N^q \tag{2}$$

$$\sum_{i=1}^{N^q} y_{ij} \leq 1, \qquad 1 \leq j \leq N \tag{3}$$

$$y_{ij} + y_{kl} \leq 1, \quad 1 \leq i<k \leq N^q, 1 \leq l<j \leq N \tag{4}$$

In the objective function given by Equation 1, $\zeta(t^q_{ik}, t_{jl})$ represents the scoring function that scores the matching of $t^q_{ik} \in T^q$ with $t_{jl} \in T$. Constraints 2 and 3 ensure that each SSE in one tableau is matched with at most one SSE in the other. Constraint 4 ensures that the matching preserves the order of the SSEs.

There are multiple ways in which the scoring function ζ can be calculated. A simple way will be to evaluate $\zeta(t^q_{ik}, t_{jl}) = 2$ if $t^q_{ik} \equiv t_{jl}$, 1 if $t^q_{ik} \simeq t_{jl}$, -2 otherwise. By $t^q_{ik} \simeq t_{jl}$ we allow the matching of entries in two tableaux entries that differ by one symbol, for example, OS and OT. Alternatively, more discrimination can be incorporated into the scoring function by using the orientation angles instead of tableau entries as arguments to ζ. The scoring function can be defined as

$$\zeta(\omega^q_{ik}, \omega_{jl}) = 45 - \Delta\omega \tag{5}$$

where $\Delta\omega = \min\{|\overline{\omega}^q_{ik} - \overline{\omega}_{jl}|, 360 - (|\overline{\omega}^q_{ik} - \overline{\omega}_{jl}|)\}$, and all angles are taken to have values between 0° and 360°.

The above quadratic program is modelled using `CPLEX` and solved using its inbuilt quadratic constraint solver to extract the maximal-common subtableau. The optimal value of $f(y)$ can be used as a measure of the similarity between tableaux.

5.2 Integer linear programming-based extraction of maximal-common subtableau

The QIP described in Section 5.1 can be formulated as an integer linear program by introducing Boolean variables x_{ijkl}, $1 \leq i, k \leq N^q$, $1 \leq j, l \leq N$. Let $x_{ijkl} = 1$ when ith and kth SSE $\in \mathcal{P}^q$ are matched with jth and lth SSE $\in \mathcal{P}$.

Using the notation introduced in Section 5.1, we have

$$x_{ijkl} = y_{ij} \wedge y_{kl} \quad \forall i, j, k, l \quad \textbf{s.t.} 1 \leq i, k \leq N^q, 1 \leq j, l \leq N.$$

The ILP can then be formulated as follows.

$$\text{maximize} \quad \sum_{1 \leq i,k \leq N^q, 1 \leq j,l \leq N} \zeta(t^q_{ik}, t_{jl}) x_{ijkl}, \tag{6}$$

subject to

$$\sum_{j=1}^{N} y_{ij} \leq 1, \qquad 1 \leq i \leq N^q \tag{7}$$

$$\sum_{i=1}^{N^q} y_{ij} \leq 1, \qquad 1 \leq j \leq N \tag{8}$$

$$y_{ij} + y_{kl} \leq 1, \qquad 1 \leq i<k \leq N^q, 1 \leq l<j \leq N \tag{9}$$

$$x_{ijkl} \leq y_{ij} \qquad 1 \leq i, k \leq N^q, 1 \leq j, l \leq N. \tag{10}$$

$$x_{ijkl} \leq y_{kl} \qquad 1 \leq i, k \leq N^q, 1 \leq j, l \leq N. \tag{11}$$

$$y_{ij} + y_{kl} \leq x_{ijkl} + 1 \qquad 1 \leq i<k \leq N^q, 1 \leq j<l \leq N \tag{12}$$

The ILP objective given by Equation 6 is equivalent to the QIP objective given by Equation 1 because $x_{ijkl} \equiv y_{ij} \times y_{kl}$. Constraints 10 and 11 ensure that the value of any x_{ijkl} cannot exceed that of y_{ij} and that of y_{kl}. Constraint 12 ensures that the values of x_{ijkl} is pushed to 1 when both y_{ij} and y_{kl} are 1. While the ILP objective (Equation 6) can be relied on to push x_{ijkl} values to 1, explicitly including this constraint will allow the ILP to converge faster to the optimal solution. Constraints 7–9 have already been described in Section 5.1.

6 RAPID COMPARISON OF TWO TABLEAUX

Although our QIP and ILP formulations give answers that are exact (although possibly not unique), they are slow to compare a query against the entire library of protein domains. To overcome this limitation, we use a simple dynamic programming (DP) method that compares two tableaux with a speed that allows the comparison of a query protein against the entire structural library in seconds. We call this method `TableauSearch`.

The aim of this method is to use an *alignment*-like approach (Needleman and Wunsch, 1997) to compare the SSE strings of two proteins, based on a scoring function derived from their tableaux. We first construct a $N^q \times N$ scoring matrix $\zeta = (\zeta_{ij})_{1 \leq i \leq N^q, 1 \leq j \leq N}$ by comparing every row in T^q with every row in T, while treating each row in any tableau as the sequence of its elements. To get a score of comparison of any ith SSE $\in \mathcal{P}^q$ ($1 \leq i \leq N^q$) with any jth SSE $\in \mathcal{P}$ ($1 \leq j \leq N$), we use a standard dynamic programming method (Bellman, 2002, Needleman and Wunsch, 1997). For a sensitive comparison, we make use the angular profiles from the orientation matrices rather than the double-encoded symbols. Each row-row comparison is an alignment of angular profiles in those rows. Equation 5 gives the score of a matching of any two orientation angles. A constant (i.e. length independent) gap penalty is fixed at -20. The optimal score of alignment of row–row angular profiles, using the above alignment parameters gives ζ_{ij} ($\forall\ 1 \leq i \leq N^q$, $1 \leq j \leq N$).

Once the scoring matrix ζ is calculated, a second tier of DP is performed to align the SSEs strings corresponding to $\mathcal{P}^q$ and $\mathcal{P}$ respectively using ζ (using the same gap penalty of -20). The optimal alignment score gives the final score of comparison. If the score is greater than 30% of the score of comparison of query with itself, it is treated as a 'hit'. We find that this threshold gives good discrimination.

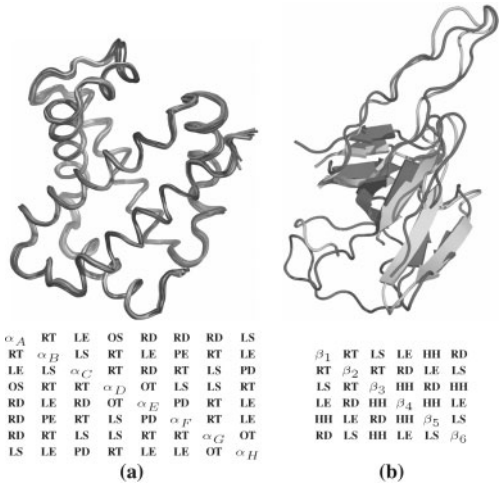
7 RESULTS

7.1 Constant-time methods

Identical or very closely related folding patterns can be extracted almost instantaneously using the `MD5` hash of a query tableau. For example, many globin proteins in the ASTRAL SCOP database have tableaux identical to the tableau of haemoglobin, Human α-chain (SCOP ID: `d1hhoa_`) (Fig. 3a). Although the eukaryotic globin fold is a good example of a large conserved core, in many cases when the structures diverge, the changes appear mostly in the loop regions, while the secondary structural core remains intact. Figure 3b shows the superposition of two Chaperonin-10 (GroES) proteins from *Mycobacterium tuberculosis* (SCOP domain: `d1hx5d_`) and *Thermus thermophilus* (SCOP domain: `d1we3s_`). Although the loop regions

changed considerably, the secondary structural geometries are conserved, resulting in identical tableaux.

Since all the tableaux of SCOP domains are precomputed, the extraction of identical and closely-related protein domains is achieved in constant-time after the tableau of the query protein is computed. An average run using these methods takes <1 s on a typical PC.



α_A	RT	LE	OS	RD	RD	RD	LS
RT	α_B	LS	RT	LE	PE	RT	LE
LE	LS	α_C	RT	RD	RT	LS	PD
OS	RT	RT	α_D	OT	LS	LS	RT
RD	LE	RD	OT	α_E	PD	RT	LE
RD	PE	RT	LS	PD	α_F	RT	LE
RD	RT	LS	LS	RT	RT	α_G	OT
LS	LE	PD	RT	LE	LE	OT	α_H

(a)

β_1	RT	LS	LE	HH	RD
RT	β_2	RT	RD	LE	LS
LS	RT	β_3	HH	RD	HH
LE	RD	HH	β_4	HH	LE
HH	LE	RD	HH	β_5	LS
RD	LS	HH	LE	LS	β_6

(b)

Fig. 3. (**a**) Superposition of around 40 different globins (top frame) sharing identical tableau (bottom frame) with that of haemoglobin, Human α-chain (SCOP domain: `d1hhoa_`). (**b**) Chaperonin-10 (GroES) proteins from *Mycobacterium tuberculosis* (SCOP domain: `d1hx5d_`, shown in yellow) and *Thermus thermophilus* (SCOP domain: `d1we3s_`, shown in blue) share same GroES-like fold containing a conserved core of secondary structural elements as shown by their superposition (top frame), resulting in identical tableaux (bottom frame).

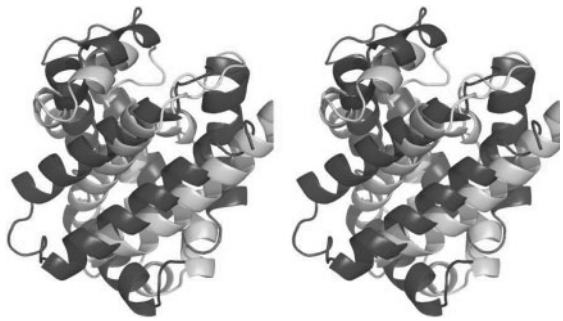
7.2 Comparison using QIP/ILP methods

The QIP and ILP methods described in Section 5 are able to detect subtle similarities between two tableaux. The result of a QIP and ILP based tableau comparison is the extraction of a maximally similar subtableau. The comparison of *Diphtheria* toxin protein from *Corynebacterium diphtheriae* (SCOP ID: `d1xdtt3`) with Human α-haemoglobin (SCOP ID: `d1hhoa_`) is a good example to illustrate the power of QIP/ILP methods. The middle domain of *Diphtheria* toxin has a globin-like fold. When the tableau of *Diphtheria* toxin's middle domain, was compared with that of Human α-haemoglobin, the QIP/ILP methods extracted of a subtableau which is consistent with the full structural alignment of the two proteins generated using MUSTANG (Konagurthu *et al.*, 2006) (Fig. 4).

In general, the QIP method converges much faster to an optimal solution than ILP. In the above example, QIP extracted the maximally-similar subtableaux in 9 s, while ILP did the same in 66 s.

7.3 TableauSearch

QIP/ILP methods, although very powerful to mine similarities between tableaux, are slow to perform a full database search. The method we proposed in Section 6 allows a rapid comparison of the entire ASTRAL SCOP domain database (containing around 75 632 proteins), while retaining its



(**a**)

α_A	RT	LE	OS	RD	RD	RD	LS
RT	α_B	LS	RT	LE	PE	RT	LE
LE	LS	α_C	RT	RD	RT	LS	PD
OS	RT	RT	α_D	OT	LS	LS	RT
RD	LE	RD	OT	α_E	PD	RT	LE
RD	PE	RT	LS	PD	α_F	RT	LE
RD	RT	LS	LS	RT	RT	α_G	OT
LS	LE	PD	RT	LE	LE	OT	α_H

(**b**)

α_A	LE	RT	LS	LS	OS	RD	RT	PE	OT
LE	α_B	OT	RD	LE	OS	RD	PE	PE	RT
RT	OT	α_C	LS	RT	LE	LS	PD	OT	LE
LS	RD	LS	α_D	RT	RT	OT	RT	LS	RD
LS	LE	RT	RT	α_E	RT	RT	LS	LE	RT
OS	OS	LE	RT	RT	α_F	LS	PE	PD	RD
RD	RD	LS	OT	RT	LS	α_G	LS	RT	LE
RT	PE	PD	RT	LS	PE	LS	α_H	OT	LE
PE	PE	OT	LS	LE	PD	RT	OT	α_I	OT
OT	RT	LE	RD	RT	RD	LE	LE	OT	α_J

(**c**)

α_A	RT	Ls	lS	RD	Rt	pe	ot
RT	α_C	LS	RT	Ls	Pd	oT	LE
Ls	LS	α_D	RT	ot	RT	LS	rD
lS	RT	RT	α_E	rT	LS	Le	RT
RD	Ls	ot	rT	α_G	ls	RT	LE
Rt	Pd	RT	LS	ls	α_H	oT	LE
pe	oT	LS	Le	RT	oT	α_I	OT
ot	LE	rD	RT	LE	LE	OT	α_J

(**d**)

Fig. 4. (**a**) MUSTANG-generated superposition (in stereo) of *Diphtheria* toxin protein, middle domain from *Corynebacterium diphtheriae* (SCOP ID: `d1xdtt3`) in brown, with haemoglobin, α-chain from Human (SCOP ID: `d1hhoa_`) in blue. (**b**) The tableau corresponding to haemoglobin. (**c**) The tableau corresponding to *Diphtheria* toxin. (**d**) The subtableau of *Diphtheria* toxin which is maximally-similar to the tableau of haemoglobin was extracted using QIP and ILP methods, in conformity with the structural alignment by MUSTANG. The dissimilar characters in the subtableau are shown in lower-case to allow a convenient comparison.

sensitivity to identify distantly-related folding patterns. We illustrate the speed and accuracy of TableauSearch on eight classic structural patterns defined by TOPS (Michalopoulos *et al*., 2004). One query per folding pattern was selected to test TableauSearch. These queries were the top-most hits returned by TOPS when a 'SCOP all' search was performed on the folding pattern. Table 1 contains the list of queries.

Figure 5 shows the multiple superpositions of top 50 significant hits for various queries listed in Table 1. The full list of hits for all the queries can be found at http://hollywood.bx.psu.edu/TabSearch/supl.html. The superpositions clearly show the quality of hits returned using TableauSearch. Table 1 also shows the time taken by various queries to search the entire library of tableaux, corresponding to the protein domains available in ASTRAL SCOP 1.71 domain database.

Table 1. Eight protein structural folding patterns, their corresponding SCOP domain identifiers used as queries to test TableauSearch, and the time taken (on a typical single processor PC) for these queries to perform an entire ASTRAL SCOP database search consisting of 75 632 domains

Fold	SCOP domain ID	Time
Greek-key	d1hr6rb_	65 s
NAD-binding fold	d1f6dc_	221 s
Immunoglobulin	d1ae6h1	24 s
Plait	d1bhne_	49 s
Jelly Roll	d2ph1b1	104 s
TIM-Barrel	d1tima_	140 s
Key-Barrel	d1tttb1	18 s
Ubiquitin-Roll	d1ubq__	66 s

Elaborating on one of the examples above, we consider the Ubiquitin-like protein: d1ubi__. This $\alpha+\beta$ protein is classified as a β-Grasp (ubiquitin-like) fold in SCOP, containing a β-β-α-β-β structural core. The β-Grasp fold in SCOP contains 12 structural superfamilies. TableauSearch search using d1ubi__ returned 351 significant domain hits (out of 75 632 domains) across 17 superfamilies in SCOP. Out of 351 significant hits, 322 corresponded to domains classified as β-Grasp in SCOP. All the 12 SCOP β-Grasp superfamilies were identified by TableauSearch. This suggests that TableauSearch is sensitive to identify even distant similarities.

Often, a query folding pattern forms only a small part of a much larger domain. For example, Shi *et al*. (2007) analysed the β-Grasp-containing proteins and classified them into three categories: β-grasp core, gregarious folds and structural drifts. The latter two classifications contain proteins with a (not necessarily complete) β-Grasp pattern subsumed within a larger domain. The default parameters of TableauSearch will fail to recognize such subsumed patterns. Dropping the terminal gap-penalties in both the dynamic programming phases of TableauSearch will facilitate their recognition. Using the d1ubi__ example, without the terminal gap-penalties, we identified most of the SCOP folds which contain a β-grasp pattern, listed by Shi *et al*. (2007). (The missing ones are: BtrG-like and Knottins.)

Recently, the crystal structure of a bacterial MACPF protein, Plu-MACPF from *Photorhadus luminescens* was reported (PDB ID: 2qp2) (Rosado *et al*., 2007). One of the domains of this MACPF protein shares some structural similarity with

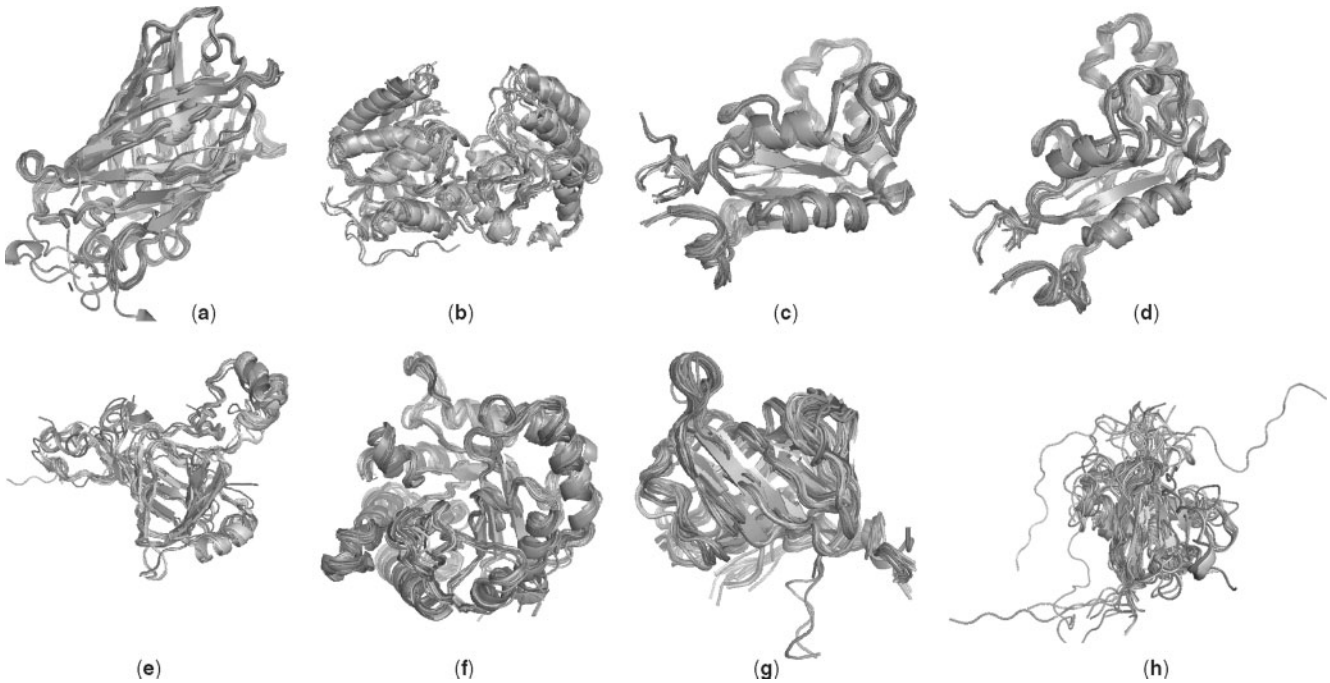


Fig. 5. Superposition of top 50 hits returned by TableauSearch on queries corresponding to eight classic folding patterns (**a**) Greek-Key, (**b**) NAD-binding fold, (**c**) Immunoglobulin, (**d**) Plait, (**e**) Jelly Roll, (**f**) TIM-Barrel, (**g**) Key-Barrel and (**h**) Ubiquitin-Roll

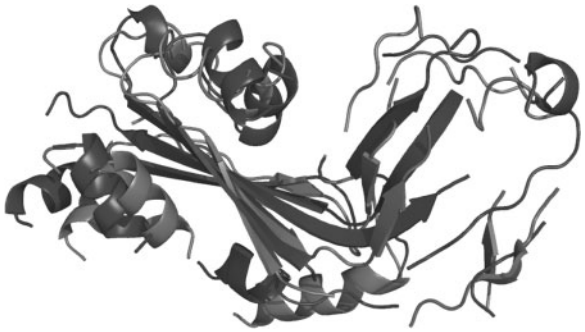


Fig. 6. Superposition of the core of MACPF protein and the bacterial perfringolysin. MUSTANG was used to superpose the two protein structures.

pore-forming cholesterol-dependent cytolysins from Gram-positive bacteria (Rosado *et al*., 2007). We queried `TableauSearch` with this newly found structure. Consistent with the report in the article, the highest scoring hit corresponds to Perfringolysin from *Clostridium perfringens* (SCOP ID: `d1m3ic_`). Popular residue-level structural alignment programs, such as DALI (Holm and Sander, 1993), fail to find any structural similarity between these two protein structures (see Rosado *et al*. (2007), supplementary data). Figure 6 shows the superposition of the core of the two protein structures. This further shows the sensitivity of TableauSearch in detecting such similarities.

8 CONCLUSION

We have shown the effectiveness of the tableau representation of protein structures to perform structural database searches. We described several methods to mine similarities between tableaux of proteins. We built `TableauSearch` which allows the retrieval of similar folding patterns from ASTRAL SCOP 1.71 domain database in seconds on an ordinary PC. We showed that `TableauSearch` is sensitive enough to detect distant structural similarities. Moreover, such rapid and sensitive methods are extremely useful as filters for slower, higher-resolution methods such as QIP and ILP described in this article, or full structural comparison methods such as MUSTANG. The methods described in this article will allow the automatic classification of newly solved protein structures. These methods will also aid the analysis of protein domains which share common folding patterns.

Conflict of Interest: none declared.

REFERENCES

Abagyan,R.A. and Maiorov,V.N. (1988) A simple qualitative representation of polypeptide chain folds: comparison of protein tertiary structures. *J. Biomol. Struct. Dyn.*, **5**, 1267–1279.

Artymiuk,P.J. *et al*. (1992) Similarity searching in databases of three-dimensional molecules and macromolecules. *J. Chem. Inf. Comput. Sci.*, **32**, 617–630.

Artymiuk,P.J. *et al*. (1992) Three-dimensional structural resemblance between leucine aminopeptidase and carboxypeptidase A revealed by graph-theoretical techniques. *FEBS Lett.*, **303**, 48–52.

Bellman,R. (1957) *Dynamic Programming*. Princeton University Press, Princeton, NJ.

Berman,H.M. *et al*. (2002) The Protein Data Bank. *Acta Crystallogr. D Biol. Crystallogr.*, **58** (Pt 6 No 1), 899–907.

Chandonia,J.M. *et al*. (2004) The ASTRAL compendium in 2004. *Nucleic Acids Res.*, **32** (Database issue).

Grindley,H.M. *et al*. (1993) Identification of tertiary structure resemblance in proteins using a maximal common subgraph isomorphism algorithm. *J. Mol. Biol.*, **229**, 707–721.

Harrison,A. *et al*. (2003) Recognizing the fold of a protein structure. *Bioinformatics*, **19**, 1748–59.

Holm,L. and Sander,C. (1993) Protein structure comparison by alignment of distance matrices. *J. Mol. Biol.*, **233**, 123–138.

Kabsch,W. and Sander,C. (1983) Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, **22**, 2577–2637.

Kamat,A.P. and Lesk,A.M. (2007) Contact patterns between helices and strands of sheet define protein folding patterns. *Proteins: Struct. Funct. Bioinform.*, **66**, 869–876.

Koch,I. *et al*. (1996) An algorithm for finding maximal common subtopologies in a set of protein structures. *J. Comput. Biol.*, **3**, 289–306.

Konagurthu,A.S. *et al*. (2006) MUSTANG: a multiple structural alignment algorithm. *Proteins: Struct. Funct. Bioinform.*, **64**, 559–574.

Lesk,A.M. and Chothia,C. (1980) How different amino acid sequences determine similar protein structures: the structure and evolutionary dynamics of the globins. *J. Mol. Biol.*, **136**, 225–270.

Lesk,A.M. (1995) Systematic representation of protein folding patterns. *J. Mol. Graphics*, **13**, 159–164.

Lo Conte,L. *et al*. (2000) SCOP: a structural classification of proteins database. *Nucleic Acids Res.*, **28**, 257–259.

Madej,T. *et al*. (1995) Threading a database of protein cores. *Proteins*, **23**, 356–369.

Michalopoulos,I. *et al*. (2004) TOPS: an enhanced database of protein structural topology. *Nucleic Acid Res.*, **32**, D251–D254.

Mizuguchi,K. and Go,N. (1995) Comparison of spatial arrangements of secondary structural elements in proteins. *Protein Eng.*, **8**, 353–362.

Needleman,S.B. and Wunsch,C.D. (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.*, **48**, 443–453.

Orengo,C.A. (1997) CATH–a hierarchic classification of protein domain structures. *Structure*, **5**, 1093–1108.

Orengo,C.A. and Taylor,W.R. (1990) A rapid method for protein structure alignment. *J. Theor. Biol.*, **147**, 517–551.

Papadimitriou,C.H. and Steiglitz,K. (1998) *Combinatorial Optimization: Algorithms and Complexity*. Dover Publications, New York.

Rivest,R. (1992) The MD5 message digest algorithm, RFC 1321. MIT and RSA Data Security, Inc.

Rosado,C.J. *et al*. (2007) A common fold mediates vertebrate defense and bacterial attack. *Science*, **317**, 1548–1551.

Rufino,S. and Blundell,T. (1994) Structure-based identification and clustering of protein families and super-families. *J. Computer Aided Mol. Design*, **8**.

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.

Shi,S. *et al*. (2007) Searching for three-dimensional secondary structural patterns in proteins with ProSMoS. *Bioinformatics*, **23**, 1331–1338.