Genome Informatics 21: 65-76 (2008)

BSAlign: A RAPID GRAPH-BASED ALGORITHM FOR DETECTING LIGAND-BINDING SITES IN PROTEIN STRUCTURES

ZEYAR AUNG azeyar@i2r.a-star.edu.sg
JOO CHUAN TONG jctong@i2r.a-star.edu.sg

*Institute for Infocomm Research, A*STAR (Agency for Science, Technology and Research), 1 Fusionopolis Way, #21-01 Connexis, Singapore 138632*

Detection of ligand-binding sites in protein structures is a crucial task in structural bioinformatics, and has applications in important areas like drug discovery. Given the knowledge of the site in a particular protein structure that binds to a specific ligand, we can search for similar sites in the other protein structures that the same ligand is likely to bind. In this paper, we propose a new method named "BSAlign" (**B**inding **S**ite **Align**er) for rapid detection of potential binding site(s) in the target protein(s) that is/are similar to the query protein's ligand-binding site. We represent both the binding site and the protein structure as graphs, and employ a subgraph isomorphism algorithm to detect the similarities of the binding sites in a very time-efficient manner. Preliminary experimental results show that the proposed BSAlign binding site detection method is about 14 times faster than a well-known method called SiteEngine, while offering the same level of accuracy. Both BSAlign and SiteEngine achieve 60% search accuracy in finding adenine-binding sites from a data set of 126 proteins. The proposed method can be a useful contribution towards speed-critical applications such as drug discovery in which a large number of proteins are needed to be processed. The program is available for download at: http://www1.i2r.a-star.edu.sg/~azeyar/BSAlign/.

Keywords: protein structure; ligand-binding site; efficient binding site detection; subgraph isomorphism; adenine-binding sites.

1. Introduction

Proteins are the physical basis of life, and perform a number of vital functions such as storage, structural lattice, movement, transport, signaling, immunity, catalysis in metabolism, etc. A ligand is a specific compound that binds to a particular receptor protein to form a complex. It can inhibit, promote, or alter the function of the receptor protein. A ligand can either be another protein or a non-protein small molecule. Drugs are examples of small molecule ligands.

A ligand-binding site is a region in a receptor protein structure to which a ligand binds. Binding site detection is a task in which, given the knowledge of the binding site in a particular protein structure a specific ligand binds to, we detect in the other protein structure(s) for the site(s) with the similar structural and physicochemical characteristics, where the same ligand is likely to bind — as illustrated in Figure 1.

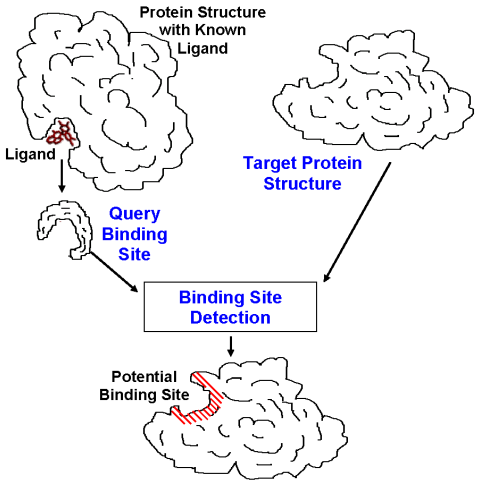

Fig. 1. Detection of a potential binding site similar to the query binding site.

This is a crucial task in structural bioinformatics, and has important applications in the area of drug discovery. In particular, binding site detection is a very useful mechanism for identifying the new drug targets and developing the targeted drug leads like inhibitors [20]. In addition to drug discovery, binding site detection is also useful for protein function prediction [14].

In this paper, we propose a new method named "BSAlign" (**B**inding **S**ite **Align**er) that detects the potential site(s) in a target protein that is/are similar to the query binding site where a specific ligand is know to bind. The method is designed to compare a query site against the similar site(s) in a single target protein, but can easily be adapted to search for potential sites in multiple target proteins.

The BSAlign method represents both the query binding site and the target protein structure as graphs. The graph representation scheme that we use captures information on both the geometrical conformations and the physicochemical properties of amino acid residues in the query and the target. Then, the method applies a subgraph isomorphism algorithm to find the maximum common subgraph(s) of the input graphs. The subgraph isomorphism problem can be effectively solved by transforming the two input graphs into an edge-product graph, and finding the maximum clique(s) or the fully-connected subgraph(s) in the edge product graph [9, 12]. From the maximum clique(s), the list(s) of maximally matching residue pairs is/are extracted. After that, those list(s) of matching residue pairs is/are refined with respect to a scoring function in order to yield the final list of optimally matching/aligned residue pairs. Depending on the size and density of the input graphs, the method automatically tunes the matching criteria of the graphs' vertices and edges on the fly so as to avoid a lengthly subgraph isomorphism process.

We tested our method by detecting the adenine-binding sites in a data set of 126 protein structures. The experimental results show that BSAlign can detect the potential binding sites for adenine-containing ligands efficiently and effectively. BSAlign is compared against another state-of-the-art binding site detection method named SiteEngine [20]. It is observed that BSAlign is 14 times as fast as SiteEngine while providing as good accuracy (60%) as SiteEngine. Since speed is a crucial factor for applications like drug discovery, which involve large quantities of ligands, ligand-binding proteins and potential target proteins [3], the efficiency of our proposed BSAlign method can be an important contribution towards such speed-critical applications.

2. Related Works

The problem of binding site detection is related to that of protein substructure alignment since both involve identifying a region similar to the query substructure in the target protein. However, the generic substructure alignment methods such as [5, 7, 19] cannot be effectively used for binding site detection, because they take only the geometrical properties of residues into account, but not their physicochemical attributes, which are essential in identifying the ligand-binding residues.

A number of algorithms dedicated to binding site detection/prediction have been proposed. The methods such as [1, 11, 14] predict potential binding sites on the surfaces of proteins without an *a priori* knowledge of a similar binding site. On the other hand, the methods such as [4, 8, 17, 18, 20] detect the target protein's potential binding site(s) which is/are similar to the query binding site.

ASSAM [4] represents residue side-chains as pseudo-atoms, and performs subgraph isomorphism to detect the side-chain patterns common to a set of binding sites. eF-site [8] and Cavbase [18] represent a binding sites as a set of detailed surface points and pseudocenters (selected atoms) in residues respectively, and apply subgraph isomorphism to find the similar binding sites. However, given the usually large quantities of objects (surface points or pseudocenters) in a query binding site and a target protein and the complexity of the subgraph isomorphism problem, which is NP-hard [15], these methods are not time-efficient.

SiteEngine [20] represents a binding site as a set of pseudocenters (as in Cavbase [18]), and applies geometric hashing to detect the binding site similarities. Being based on the efficient geometric hashing technique, it is faster than Cavbase. However, its time efficiency is still inadequate when a large amount of query binding sites and target proteins are to be processed, as usually needed in the case of drug discovery [3].

A recently proposed method, SiteAlign [17], encodes binding sites as fixed-length cavity fingerprints, and performs a time-efficient comparison on these fingerprints. No accuracy comparison of SiteAlign with those of the other methods is available. However, in general, the accuracy of fingerprint-based comparison methods tend to be lower than those of detailed comparison methods [21].

Our objective is to overcome the shortcomings, either in terms of time efficiency and accuracy, of the abovementioned methods. In order to achieve a better time efficiency, we adopt a residue-based approach, as opposed to the finer-grained approaches [8, 18, 20], which use sub-residue information like surface points or pseudocenters. On the other hand, in order to achieve the same level of accuracy as those finer-grained methods, we carefully design our residue-based graph representation scheme to encompass enough geometric and physicochemical information, and employ subgraph isomorphism for a detailed graph comparison. Our preliminary experimental results show that we have achieved our objective, and come up with a solution that is much faster than the fastest of the finer-grained methods, namely SiteEngine [20], while maintaining the same level of accuracy.

3. The BSAlign Method

3.1. *Graph Representation*

The input to the BSAlign algorithm are the query binding site and the entire target protein structure. We can define a binding site as a set of residues that are interacting with the ligand in question. A residue is considered to be interacting with the ligand if it is within 5Å radius from the ligand [13].

Both the query binding site and the target protein structure can be represented as graphs. Since the sequence order of residues is irrelevant in comparing and detecting binding sites [6], the graph representation, which is sequence-order independent, is best suited for our purpose. We use a residue-based graph representation scheme which captures information on both geometrical and physicochemical properties of the amino acid residues. Each residue is encoded as a vertex in the graph. Two vertices, representing two residues, are connected by an edge if these two residues are close enough to each other, i.e., the distance between their Cα atoms is less than or equal to 15Å (an empirically determined value). A vertex is characterized by a vertex label which comprises of the following attributes:

(1) Solvent accessibility of the residue as a percentage (0∼100%) (denoted as SA),
(2) Physicochemical type (non-polar, polar, aromatic, positive, or negative) of the residue (PT), and
(3) Secondary structure type (helix, sheet, or loop) of the residue (SS).

An edge connecting two vertices (residues) is characterized by an edge label comprising the following attributes:

(1) Distance between the Cα atoms of the two residues (DC) and
(2) Angle between the Cα–Cβ vectors of the two residues (AN). (A Cα–Cβ vector is an imaginary line segment connecting the Cα and the Cβ atoms of a residue.)

Among these attributes, PT, DC and AN can be derived simply from the PDB files (`http://www.rcsb.org`), and SA and SS can be obtained by using the DSSP program (`http://swift.cmbi.kun.nl/gv/dssp/`).

3.2. *Graph Similarity*

The similarity between two graphs can be determined by finding the maximum common subgraph in them. The larger the common subgraph, the more similar the two given graphs are. The maximum common subgraph problem can be solved by transforming the two input graphs into a single edge-product graph and finding the maximum clique (fully-connected subgraph) in that edge-product graph [9, 12].

3.2.1. *Edge-product Graph Construction*

Let G be a graph of any kind defined as $G = (V, E)$ where V is the set of vertices and E is the set of edges in G respectively. We can express V as $\{v_i \,|\, i = 1 \ldots |V|\}$ where $|V|$ is the number of vertices in G. Similarly, was can express E as $\{e_i \,|\, i = 1 \ldots |E|\}$ where $|E|$ is the number of edges in G. An edge e_i can in turn be expressed as $e_i = (a_i, b_i)$ where $a_i, b_i \in V$ are the two vertices connected by e_i.

An edge-product graph GP of two input graphs $G1 = (V1, E1)$ and $G2 = (V2, E2)$ is defined as $GP = (VP, EP) = (E1 \times E2)$ in which:

- The vertex set VP of the product graph consists of all the compatible edge pairs in $E1$ and $E2$. That is, $vp_i = (e1_r, e2_s)$ if:
  - $EC(e1_r, e2_s) = \text{TRUE}$, and
  - $(VC(a1_r, a2_s) = \text{TRUE} \wedge VC(b1_r, b2_s) = \text{TRUE}) \vee$ $(VC(a1_r, b2_s) = \text{TRUE} \wedge VC(b1_r, a2_s) = \text{TRUE})$
- There exists an edge between two vertices $vp_i = (e1_r, e2_s)$ and $vp_j = (e1_t, e2_u)$ of the product graph if:
  - $(e1_r \neq e1_t) \wedge (e2_s \neq e2_u)$, and
  - Either:
    - ($e1_r$ and $e1_t$ have a common vertex $v1_{rt}$) $\wedge$ ($e2_s$ and $e2_u$ have a common vertex $v2_{su}$) $\wedge$ $(VC(v1_{rt}, v2_{su}) = \text{TRUE})$, or
    - ($e1_r$ and $e1_t$ do not have a common vertex) $\wedge$ ($e2_s$ and $e2_u$ do not have a common vertex)

The vertex compatibility function VC of the two vertices of v_i from $G1$ and v_j from $G2$ is defined as:

$$VC(v_i, v_j) = \begin{cases} \text{TRUE} & \text{if } (|v_i.SA - v_j.SA| \leq T1_{SA}) \vee \\ & \quad (\,(|v_i.SA - v_j.SA| \leq T2_{SA}) \wedge \\ & \quad (v_i.PT = v_j.PT) \wedge (v_i.SS = v_j.SS)\,) \\ \text{FALSE} & \text{otherwise} \end{cases} \tag{1}$$

where $T1_{SA}$ and $T2_{SA}$ are the two threshold values for the differences in solvent accessibility. $T1_{SA}$ is usually a very small value, and $T2_{SA}$ is a relatively larger one. The meaning of the function $VC(v_i, v_j)$ is that the two vertices (residues) v_i and v_j are regarded as compatible if either their solvent accessibility percentages

are very close, or their accessibility percentages are close enough, and both of their physicochemical types and secondary structure types are respectively the same.

Similarly, the edge compatibility function EC of the two edges e_i from $G1$ and e_j from $G2$ is defined as:

$$EC(e_i, e_j) = \begin{cases} \text{TRUE} & \text{if } ((|e_i.DC - e_j.DC| \leq T_{DC}) \wedge \\ & \quad (|e_i.AN - e_j.AN| \leq T_{AN})) \\ \text{FALSE} & \text{otherwise} \end{cases} \tag{2}$$

where T_{DC} and T_{AN} are the threshold values for the differences in Cα–Cα distances and (Cα–Cβ)–(Cα–Cβ) angles of the two residues respectively. The function $EC(e_i, e_j)$ means that the two edges e_i and e_j are compatible if their distance and angle values in one edge are not very different from their counterparts in the other.

After we have constructed the edge-product graph, the next step is to detect the maximum clique(s) in it. Since maximum clique detection is an NP-hard problem [15], this will be the most time-consuming step in the BSAlign algorithm. In order to reduce the time taken for this step, we have to keep the size of the edge-product graph reasonably small. So, if required, we iterate the edge-product graph construction process up to 5 rounds with stricter threshold values for $T1_{SA}$, $T2_{SA}$, T_{DC} and T_{AN} at each time. We stop the iteration when number of edges in the edge-product graph becomes less than 1,000,000. For the first round, we use $T1_{SA} = 0.05$, $T2_{SA} = 0.30$, $T_{DC} = 2.0$ and $T_{AN} = 30$. For the second round, we use $T1_{SA} = 0.04$, $T2_{SA} = 0.25$, $T_{DC} = 1.5$ and $T_{AN} = 25$, and so on. For the last (fifth) round, we use $T1_{SA} = 0.01$, $T2_{SA} = 0.10$, $T_{DC} = 0.01$ and $T_{AN} = 10$. All of these values are empirically determined.

3.2.2. *Maximum Clique Detection*

After the final edge-product graph is obtained, we use the Cliquer program [15] to detect the maximum clique(s) in it. Cliquer is an implementation of a branch-and-bound maximum clique detection algorithm [16]. A brief description of the Cliquer algorithm as described in [15] is as follows:

The algorithm assume some order for the vertices $V = \{v_1, v_2, \ldots, v_{|V|}\}$. Let $S_i = \{v_1, v_2, \ldots, v_i\} \subseteq V$. The function $c(i)$ is defined to be the size of the maximum clique in the subgraph induced by S_i. Obviously, for every $i = 1, \ldots, |V| - 1$, we have either $c(i+1) = c(i)$ or $c(i+1) = c(i) + 1$. Moreover, $c(i+1) = c(i) + 1$ if and only if there exists a clique in S_{i+1} of size $c(i) + 1$ that includes vertex v_{i+1}. Cliquer calculates the values of $c(i)$ starting from $c(1) = 1$ up, and stores the values found. This enables a pruning strategy not found in older clique detection algorithms. Namely, when Cliquer is calculating $c(i + 1)$ (that is searching for a clique of size $c(i)+1$ within S_{i+1}), and it has formed a clique W and is considering adding vertex v_j, it can prune the search if $|W| + c(j) \leq c(i)$. Trivially, if it finds a clique of size $c(i)+1$, it can prune the whole search and start calculating $c(i+2)$. When searching for all maximum cliques, Cliquer first determines the size of the maximum cliques, then starts the search again at the suitable position.

3.2.3. *Matching Residue Pair Generation*

The maximum clique(s) produced by Cliquer is/are mapped back into the list(s) of matching vertex pairs by using the Hungarian maximal assignment algorithm [10]. From the list of matching edge pairs, the algorithm produces the maximum possible number of matching vertex (residue) pairs — as exemplified in Figure 2. The implementation of the Hungarian algorithm is adapted from the one described in [22].

Matching Edge Pairs				Matching Vertex Pairs		
(query)		(target)		(query)		(target)
1, 2	–	53, 55		1	–	53
1, 8	–	51, 53		2	–	55
2, 3	–	55, 57	⇒	8	–	51
3, 4	–	57, 60		3	–	57
4, 5	–	58, 60		4	–	60
7, 9	–	54, 56		5	–	58
				7	–	54
				9	–	56

Fig. 2. An example of mapping matching edge pairs into matching vertex pairs.

3.3. *Refinement and Scoring*

The two sets of matching (aligned) residue pairs are tested for their actual structural similarity using the root mean square deviation (RMSD) criterion. RMSD is calculated by superimposing the set of Cα atoms of the aligned residues in the query binding site onto their counterparts in the target protein. The smaller the RMSD, the more structurally similar the two sets of aligned residues are. However, in some cases, the RMSD values are quite large if all of the aligned residue pairs are taken into account. Therefore, we iteratively refine the initial list of aligned residue pairs by removing at each step the pair that is least fitting when superimposed. But, on the other hand, we should not remove too many pairs, because the alignment result will not be very meaningful if number of aligned residues is too small. In other words, we must balance the RMSD value the number of aligned residues in order to get the optimal alignment results. For that, we use Alexandrov and Fischer's scoring function [2], which is defined as:

$$\text{Alignment score} = \frac{3 \times \text{No. of aligned residues}}{1 + \text{RMSD}} \tag{3}$$

The refinement of the alignment is repeated until the alignment score cannot be further increased, or until the number of aligned residues is equal to one-third of the number of residues in the original query binding site. Then, the final set of aligned residues in the target protein is reported as the potential binding site. Sometimes, there are more than one maximum clique in the edge-product graph, and consequently, more than one initial lists of aligned residues exist. In such a

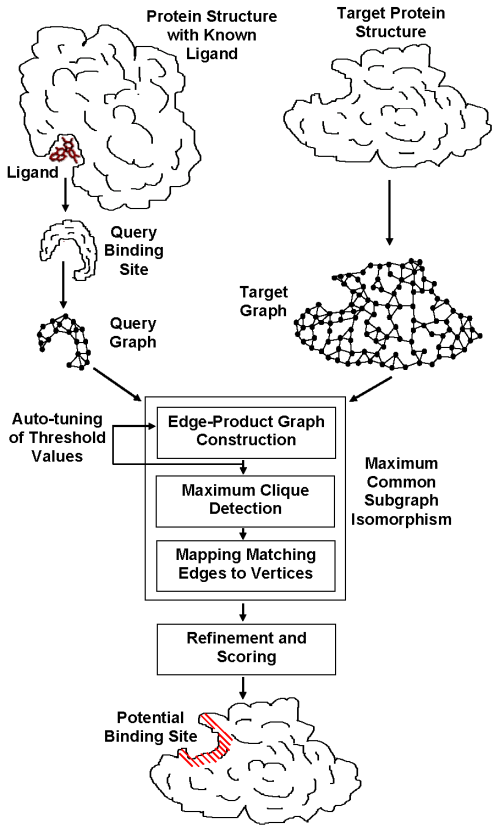

Fig. 3. Outline of the BSAlign method.

case, we refine all the available lists, and take the one that gives the highest final alignment score as the answer.

The steps taken in the BSAlign method are summarized in Figure 3.

4. Results and Discussions

Following the experiment described in [20], we test BSAlign by searching for the binding sites similar to the ATP-binding site of an adenine-binding protein "1atp" in a data set of 126 proteins listed in Table 1. The data set consists of 34 adenine-binding proteins belonging to 18 distinct SCOP Folds, and 92 proteins of other functional types from 21 distinct SCOP Folds. (SCOP – `http://scop.mrc-lmb.cam.ac.uk/scop/` is a database for structural classification of proteins. If two proteins belong to different SCOP Folds, they are very diverse in terms of their whole structures.) Adenine-binding proteins are a functional type of protein that binds to adenine-containing ligands like ATP, ANP, FAD, NAD, etc.

Table 1. The data set of 126 proteins (34 adenine-binding proteins and 92 other proteins).

Functional Type	Total	SCOP Folds	PDB IDs
Adenine-binding proteins	34	18	1a49, 1a82, 1ads, 1atp, 1ayl, 1b4v, 1b8a, 1bx4, 1byq, 1csc, 1csn, 1e2q, 1e8x, 1f9a, 1fmw, 1g5t, 1gn8, 1hck, 1hpl, 1j7k, 1jjv, 1kay, 1kp2, 1kpf, 1mjh, 1mmg, 1nhk, 1nsf, 1phk, 1qmm, 1yag, 1zin, 2src, 9ldt
Other proteins	92	21	1a27, 1a52, 1abi, 1acb, 1alq, 1arb, 1azm, 1b56, 1b6o, 1bt5, 1cbs, 1cho, 1com, 1cqq, 1cse, 1csm, 1dbf, 1dcs, 1e6w, 1ecm, 1ela, 1elc, 1equ, 1ere, 1err, 1exm, 1fby, 1fds, 1fem, 1flj, 1fnj, 1fnk, 1ftp, 1g5y, 1ghp, 1gx9, 1hah, 1har, 1hms, 1hne, 1hsg, 1hsh, 1hwr, 1ifc, 1jd0, 1jgl, 1keq, 1kop, 1kqw, 1kzk, 1l2i, 1lhu, 1lib, 1lid, 1lie, 1lvo, 1mbm, 1mdc, 1mml, 1mu2, 1oh0, 1opa, 1opb, 1pek, 1pmp, 1ppf, 1pro, 1q2w, 1qjg, 1qkt, 1rxf, 1sbn, 1sga, 1sgc, 1tgs, 1tyr, 1vrt, 1whs, 1ysc, 1znc, 2alp, 2cbr, 2ifb, 2lbd, 2lpr, 3ert, 3prk, 3sga, 3tec, 4csm, 4sgb, 4tgl
Total	126		

4.1. *Search Accuracy*

Using the BSAlign algorithm, the query ATP-binding site of 1atp is compared with every protein structure in the data set of 126 proteins in order to detect the similar binding sites in them. Then, the found binding sites are ranked by their alignment scores (Equation 3). We assess the ranking results by using the same evaluation criterion as described in [20]. We examine the 15 top ranking binding sites, and observe that 9 out of 15 (60%) belong to the adenine-binding proteins with the ligand ATP or the other adenine-containing ones (such as ANP and AP5) — as shown in Table 2. BSAlign's accuracy performance is as good as that of SiteEngine [20], which is a finer-grained method that takes the sub-residue information (namely pseudo-centers) into account. SiteEngine also ranks 9 adenine-binding proteins among its top 15 answers. Among the two sets of 15 top ranking proteins by BSAlign and SiteEngine, 8 of them (1atp, 1csn, 2src, 1phk, 1chk, ajd0, 1mjh, and 1nsf) are common to both sets.

Now, let us study the details of the alignment results. We take the alignment result for the binding sites of the proteins 1atp and 1csn as an example. The ATP-binding site of 1atp consists of 13 residues: 50(G), 51(T), 52(G), 53(S), 54(F), 55(G), 57(V), 70(A), 122(Y), 123(V), 170(E), 171(N), and 184(D). Among these 13 residues, 10 are aligned with their counterparts in 1csn, with the RMSD of 0.48Å. The aligned residue pairs are: 50(G)–19(G), 52(G)–21(G), 53(S)–22(S), 55(G)–24(G), 57(V)–26(I), 70(A)–39(A), 123(V)–88(L), 170(E)–135(D), 171(N)–136(N), and 184(D)–154(D). It turns out that all of these 10 aligned residues in 1csn are within 5Å radius of the ligand ATP bound to the protein. The two ATP-binding sites of 1atp and 1csn are illustrated in Figure 4.

Table 2. The search result for the query binding site of the ligand "ATP" of the protein 1atp in the data set of 126 proteins.

Rank	PDB ID	Protein Name	SCOP Fold Name	Sequence Identity (%) [a]	Aligned Residues	RMSD (Å)	Align -ment Score	Ligand	Functional Type
1	1atp	cAMP-dependent PK, catalytic subunit	Protein kinase-like (PK-like)	100.0	13	0.00	39.00	ATP	Adenine-binding
2	1csn	Casein kinase-1, CK1, catalytic subunit	Protein kinase-like (PK-like)	17.0	10	0.48	20.24	ATP	Adenine-binding
3	2src	c-src protein tyrosine kinase	SH3-like barrel	13.4	11	0.97	16.74	ANP	Adenine-binding
4	1phk	gamma-subunit of glycogen phosphorylase kinase (Phk)	Protein kinase-like (PK-like)	24.2	8	0.58	15.18	ATP	Adenine-binding
5	1hck	Cyclin-dependent PK, CDK2	Protein kinase-like (PK-like)	19.5	9	1.06	13.12	ATP	Adenine-binding
6	3prk	Proteinase K	Subtilisin-like	2.5	6	1.10	8.55	MSU	other
7	1jd0	Carbonic anhydrase	Carbonic anhydrase	4.2	6	1.44	7.37	AZM	other
8	1mjh	"Hypothetical" protein MJ0577	Adenine nucleotide alpha hydrolase-like	15.4	6	1.47	7.27	ATP	Adenine-binding
9	1fnk	Chorismate mutase	Bacillus chorismate mutase-like	7.3	6	1.48	7.25	CSD	other
10	1zin	Adenylate kinase	P-loop containing nucleoside triphosphate hydrolases	10.1	6	1.53	7.13	AP5	Adenine-binding
11	1abi	Thrombin	Trypsin-like serine proteases	16.4	6	1.75	6.53	HMR	other
12	1hah	Eukaryotic proteases	Trypsin-like serine proteases	16.7	6	1.76	6.52	NAG	other
13	1kp2	Argininosuccinate synthetase	Adenine nucleotide alpha hydrolase-like	8.8	6	1.78	6.47	ATP	Adenine-binding
14	1dbf	Chorismate mutase	Bacillus chorismate mutase-like	3.7	6	1.79	6.45	SO4	other
15	1nsf	Hexamerization domain of N-ethylmalemide-sensitive fusion (NSF) protein	P-loop containing nucleotide triphosphate hydrolases	12.4	6	1.81	6.41	ATP	Adenine-binding

Note: [a] Calculated using EMBOSS Web Server (`http://www.ebi.ac.uk/emboss/align/`).

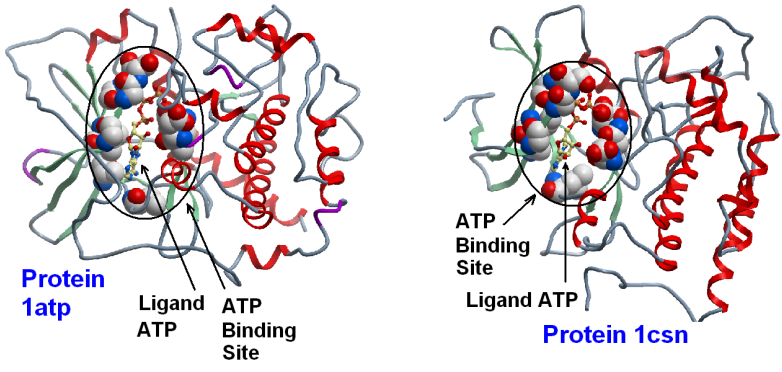

Fig. 4. ATP-binding sites of 1atp (left) and 1csn (right). Number of aligned residues = 10; RMSD = 0.48Å. The residues that involve in the alignment are shown as space-filling balls in both proteins.

4.2. *Running Time*

We compare the running times of SiteEngine and BSAlign by executing them on the same personal computer with Pentium D 3.2GHz CPU and 2GB main memory. For the aforementioned task of searching the data set of 126 proteins with the query binding site for the ligand ATP in the protein 1atp, SiteEngine takes a total of 12,010 seconds (3 hours, 18 minutes, and 10 seconds), whereas BSAlign merely takes a total of 871 seconds (14 minutes and 31 seconds). Thus, BSAlign is found to be about 14 times faster than SiteEngine while offering the same level of accuracy.

The comparable accuracy performance of the time-efficient residue-based BSAlign to that of the slower finer-grained SiteEngine can be attributed to (1) BSAlign's comprehensive graph representation scheme which captures the detailed physicochemical and geometric properties of the binding site and (2) the subgraph isomorphism process which ensures the complete matching of the two large substructures (rather than combining multiple partial matchings of the smaller substructures — as in the case of geometric hashing used by SiteEngine).

5. Conclusion

In this paper, we have presented a new ligand-binding site detection method named BSAlign, which is based on residue-based graph representation and subgraph isomorphism. Preliminary experimental results show that the method is about 14 times faster than the well-known SiteEngine method, while offering the same level of accuracy. This can be an important contribution towards the drug discovery applications where speed is critical. As a future work, BSAlign will be tested against diverse sets of protein families in order to further ascertain its accuracy and speed performances.

References

[1] Abagyan, R. and Totrov, M., High-throughput docking for lead generation, *Curr. Opin. Chem. Biol.*, 5:375–382, 2001.

[2] Alexandrov, N. N. and Fischer, D., Analysis of topological and nontopological structural similarities in the PDB: new examples with old structures, *Prot. Struct. Funct. Genet.*, 25:354–365, 1996.

[3] Alvarez, J. and Shoichet, B. (eds.), *Virtual Screening in Drug Discovery*, Taylor and Francis Ltd, 2005.

[4] Artymiuk, P. J., Poirrette, A. R., Grindley, H. M., Rice, D. W., and Willett, P., A graph-theoretic approach to the identification of three-dimensional patterns of amino acid side-chains in protein structures, *J. Mol. Biol.*, 243:327–344, 1994.

[5] Aung, Z. and Tan, K. L., MatAlign: precise protein structure comparison by matrix alignment, *J. Bioinfo. Comp. Biol.*, 4:1197–1216, 2006.

[6] Fischer, D., Wolfson, H., Lin, S. L., and Nussinov, R., Three-dimensional, sequence order-independent structural comparison of a serine protease against the crystallographic database reveals active site similarities: potential implications to evolution and to protein folding, *Protein Sci.*, 3:769–778, 1994.

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

[8] Kinoshita, K. and Nakamura, H., Identification of protein biochemical functions by similarity search using the molecular surface database eF-site, *Protein Sci.*, 12:1589–1595, 2003.

[9] Koch, I., Lengauer, T., and Wanke, E., An algorithm for finding maximal common subtopologies in a set of protein structures, *J. Comp. Biol.*, 3:289–306, 1996.

[10] Kuhn, H. W., The Hungarian Method for the assignment problem, *Nav. Res. Log. Quart.*, 2:83–97, 1955.

[11] Laurie, A. T. and Jackson, R. M., Q-SiteFinder: an energy-based method for the prediction of protein-ligand binding sites, *Bioinformatics*, 21:1908–1916, 2005.

[12] May, P., *Protein Structure Analysis using Contact Maps and Secondary Structure*, Ph.D. Dissertation, Free University of Berlin, 2007.

[13] Mohamad, S. B., Ong, A. L., and Ripen, A. M., Evolutionary trace analysis at the ligand binding site of laccase, *Bioinformation*, 2:369–372, 2008.

[14] Murga, L. F., Wei, Y., and Ondrechen, M. J., Computed protonation properties: unique capabilities for protein functional site prediction, *Genome Informatics*, 19:107–118, 2007.

[15] Niskanen, S. and Östergård, P. R. J., *Cliquer User's Guide, Version 1.0*, Technical Report T48, Communications Laboratory, Helsinki University of Technology, 2003.

[16] Östergård, P. R. J., A fast algorithm for the maximum clique problem, *Discrete Appl. Math.*, 120:195–205, 2002.

[17] Schalon, C., Surgand, J. S., Kellenberger, E., and Rognan, D., A simple and fuzzy method to align and compare druggable ligand-binding sites, *Prot. Struct. Funct. Bioinfo.*, 71:1755–1778, 2008.

[18] Schmitt, S., Kuhn, D., and Klebe, G., A new method to detect related function among proteins independent of sequence and fold homology, *J. Mol. Biol.*, 323:387–406, 2002.

[19] Shindyalov, I. N. and Bourne, P. E., Protein structure alignment by incremental combinatorial extension (CE) of the optimal path, *Protein Eng.*, 11:739–747, 1998.

[20] Shulman-Peleg, A., Nussinov, R., and Wolfson, H. J., Recognition of functional sites in protein structures, *J. Mol. Biol.*, 339:607–633, 2004.

[21] Sierk, M. L. and Pearson, W. R., Sensitivity and selectivity in protein structure comparison, *Protein Sci.*, 13:773–785, 2004.

[22] `http://www.public.iastate.edu/~ddoty/HungarianAlgorithm.html`